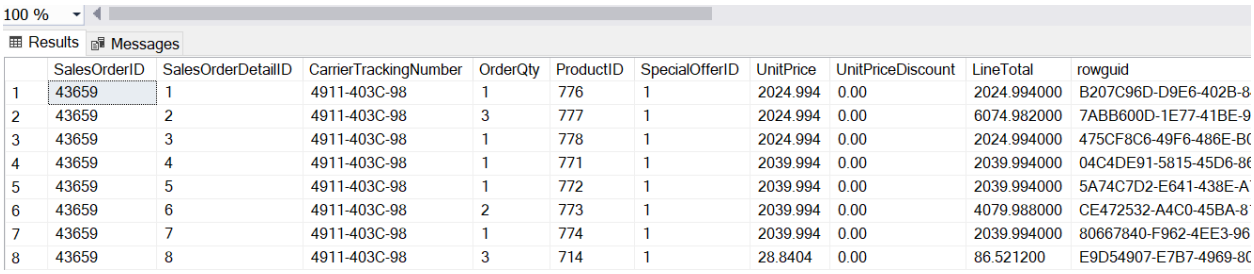
DATA PREPERATION AND CLEANING

```
-- overview database


SELECT TOP 10 *
  FROM [Sales].[SalesOrderDetail]
```

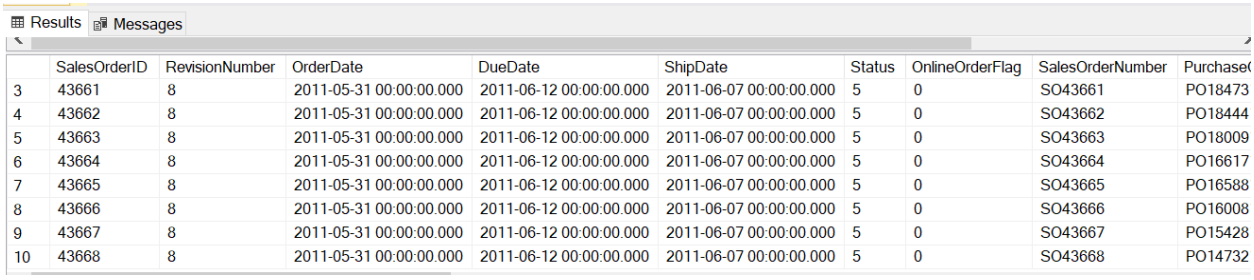


100 %

Results Messages

	SalesOrderID	SalesOrderDetailID	CarrierTrackingNumber	OrderQty	ProductID	SpecialOfferID	UnitPrice	UnitPriceDiscount	LineTotal	rowguid
1	43659	1	4911-403C-98	1	776	1	2024.994	0.00	2024.994000	B207C96D-D9E6-402B-8
2	43659	2	4911-403C-98	3	777	1	2024.994	0.00	6074.982000	7ABB600D-1E77-41BE-9
3	43659	3	4911-403C-98	1	778	1	2024.994	0.00	2024.994000	475CF8C6-49F6-486E-B(
4	43659	4	4911-403C-98	1	771	1	2039.994	0.00	2039.994000	04C4DE91-5815-45D6-8(
5	43659	5	4911-403C-98	1	772	1	2039.994	0.00	2039.994000	5A74C7D2-E641-438E-A
6	43659	6	4911-403C-98	2	773	1	2039.994	0.00	4079.988000	CE472532-A4C0-45BA-8
7	43659	7	4911-403C-98	1	774	1	2039.994	0.00	2039.994000	80667840-F962-4EE3-96
8	43659	8	4911-403C-98	3	714	1	28.8404	0.00	86.521200	E9D54907-E7B7-4969-8(

```
SELECT TOP 10 *
  FROM [Sales].[SalesOrderHeader]
```



Results Messages

	SalesOrderID	RevisionNumber	OrderDate	DueDate	ShipDate	Status	OnlineOrderFlag	SalesOrderNumber	Purchase(
3	43661	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0	SO43661	PO18473
4	43662	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0	SO43662	PO18444
5	43663	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0	SO43663	PO18009
6	43664	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0	SO43664	PO16617
7	43665	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0	SO43665	PO16588
8	43666	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0	SO43666	PO16008
9	43667	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0	SO43667	PO15428
10	43668	8	2011-05-31 00:00:00.000	2011-06-12 00:00:00.000	2011-06-07 00:00:00.000	5	0	SO43668	PO14732

```
SELECT TOP 10*
  FROM [Production].[ProductCostHistory]
```

	ProductID	StartDate	EndDate	StandardCost	ModifiedDate
3	707	2013-05-30 00:00:00.000	NULL	13.0863	2013-05-16 00:00:00.000
4	708	2011-05-31 00:00:00.000	2012-05-29 00:00:00.000	12.0278	2012-05-29 00:00:00.000
5	708	2012-05-30 00:00:00.000	2013-05-29 00:00:00.000	13.8782	2013-05-29 00:00:00.000
6	708	2013-05-30 00:00:00.000	NULL	13.0863	2013-05-16 00:00:00.000
7	709	2011-05-31 00:00:00.000	2012-05-29 00:00:00.000	3.3963	2012-05-29 00:00:00.000
8	710	2011-05-31 00:00:00.000	2012-05-29 00:00:00.000	3.3963	2012-05-29 00:00:00.000
9	711	2011-05-31 00:00:00.000	2012-05-29 00:00:00.000	12.0278	2012-05-29 00:00:00.000
10	711	2012-05-30 00:00:00.000	2013-05-29 00:00:00.000	13.8782	2013-05-29 00:00:00.000

```
-- create a temp table  #SALE

SELECT
       saleD.SalesOrderID,
          saleD.SalesOrderDetailID,
          saleD.OrderQty,saleD.ProductID,
          saleD.UnitPrice,saleH.OrderDate,
          saleH.TerritoryID,
          saleH.CustomerID,
          saleH.SalesPersonID,
          c.StandardCost

          INTO #SALE

  FROM [Sales].[SalesOrderDetail] as saleD
  JOIN [Sales].[SalesOrderHeader] as saleH
    ON saleD.SalesOrderID = saleH.SalesOrderID
  LEFT JOIN [Production].[ProductCostHistory] as c
    ON saleD.ProductID=c.ProductID
          AND saleH.OrderDate >=c.StartDate
          AND saleH.OrderDate <= COALESCE(c.EndDate,GETDATE());


SELECT TOP 10 *
  FROM #SALE
```

100 %

Results | Messages

	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	OrderDate	TerritoryID	CustomerID	SalesPersonID	StandardCost
1	43659	1	1	776	2024.994	2011-05-31 00:00:00.000	5	29825	279	1898.0944
2	43659	2	3	777	2024.994	2011-05-31 00:00:00.000	5	29825	279	1898.0944
3	43659	3	1	778	2024.994	2011-05-31 00:00:00.000	5	29825	279	1898.0944
4	43659	4	1	771	2039.994	2011-05-31 00:00:00.000	5	29825	279	1912.1544
5	43659	5	1	772	2039.994	2011-05-31 00:00:00.000	5	29825	279	1912.1544
6	43659	6	2	773	2039.994	2011-05-31 00:00:00.000	5	29825	279	1912.1544
7	43659	7	1	774	2039.994	2011-05-31 00:00:00.000	5	29825	279	1912.1544
8	43659	8	3	714	28.8404	2011-05-31 00:00:00.000	5	29825	279	31.7244
9	43659	9	1	716	28.8404	2011-05-31 00:00:00.000	5	29825	279	31.7244
10	43659	10	6	709	5.70	2011-05-31 00:00:00.000	5	29825	279	3.3963

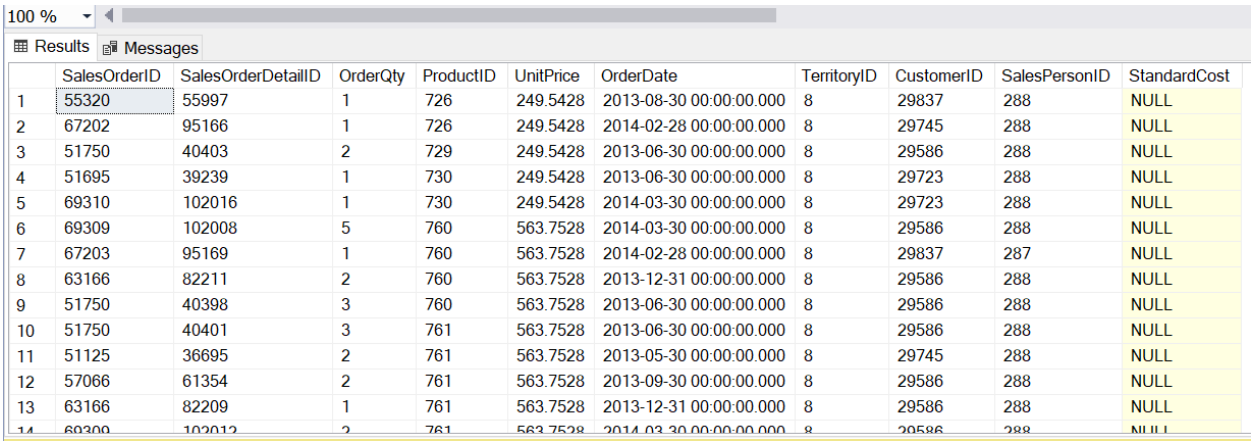
```sql
-- check missing value

SELECT count(*) - Count(SalesOrderID),
count(*) - Count(SalesOrderDetailID),
count(*) - Count(OrderQty),
count(*) - Count(OrderDate),
count(*) - Count(ProductID),
count(*) - Count(UnitPrice),
count(*) - Count(TerritoryID),
count(*) - Count(CustomerID),
count(*) - Count(SalesOrderID),
count(*) - Count(StandardCost) AS standardCostmissing
FROM #SALE;
```

100 %

Results | Messages

	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	standardCostmissing
1	0	0	0	0	0	0	0	0	64

```sql
-- show the missing value

SELECT *
  FROM #SALE
 WHERE StandardCost is NULL
 ORDER BY ProductID;
```



100 %

Results | Messages

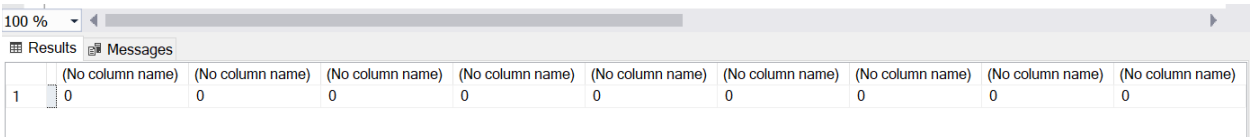
	SalesOrderID	SalesOrderDetailID	OrderQty	ProductID	UnitPrice	OrderDate	TerritoryID	CustomerID	SalesPersonID	StandardCost
1	55320	55997	1	726	249.5428	2013-08-30 00:00:00.000	8	29837	288	NULL
2	67202	95166	1	726	249.5428	2014-02-28 00:00:00.000	8	29745	288	NULL
3	51750	40403	2	729	249.5428	2013-06-30 00:00:00.000	8	29586	288	NULL
4	51695	39239	1	730	249.5428	2013-06-30 00:00:00.000	8	29723	288	NULL
5	69310	102016	1	730	249.5428	2014-03-30 00:00:00.000	8	29723	288	NULL
6	69309	102008	5	760	563.7528	2014-03-30 00:00:00.000	8	29586	288	NULL
7	67203	95169	1	760	563.7528	2014-02-28 00:00:00.000	8	29837	287	NULL
8	63166	82211	2	760	563.7528	2013-12-31 00:00:00.000	8	29586	288	NULL
9	51750	40398	3	760	563.7528	2013-06-30 00:00:00.000	8	29586	288	NULL
10	51750	40401	3	761	563.7528	2013-06-30 00:00:00.000	8	29586	288	NULL
11	51125	36695	2	761	563.7528	2013-05-30 00:00:00.000	8	29745	288	NULL
12	57066	61354	2	761	563.7528	2013-09-30 00:00:00.000	8	29586	288	NULL
13	63166	82209	1	761	563.7528	2013-12-31 00:00:00.000	8	29586	288	NULL
14	69309	102012	2	761	563.7528	2014-03-30 00:00:00.000	8	29586	288	NULL

```
-- The reason why StandardCost return to NULL for some rows is in the ProductCostHistory
the EndDate is less than the OrderDate in SaleHeader table.
-- So the solution is create a table called MaxCost to get the lasted cost for each
ProductID and after that join into the #SALE table




-- correct the null value
-- Create a table with the last price for each productid


with lastchange AS
(SELECT
       ProductID, MAx(ModifiedDate) AS lastchangedate
  FROM [Production].[ProductCostHistory]
  GRoup by ProductID)

SELECT l.ProductID,p.StandardCost

  INTO #MaxCost

  FROM [Production].[ProductCostHistory] as p
  JOIN lastchange as l
  ON     l.ProductID = p.ProductID
   AND   l.lastchangedate =p.ModifiedDate
ORDER BY l.ProductID



SELECT TOp 10 *
  FROM #MaxCost
```

Results | Messages

	ProductID	StandardCost
1	999	343.6496
2	998	343.6496
3	997	343.6496
4	996	53.9416
5	995	44.9506
6	994	23.9716
7	993	294.5797
8	992	294.5797
9	991	294.5797
10	990	294.5797

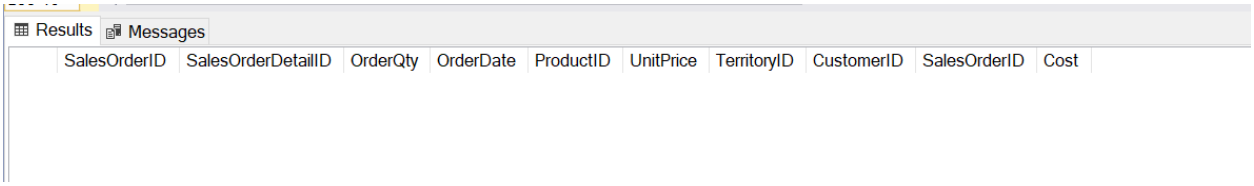
```
-- create #SALEFINAL table hen fix teh null value of #SALE table

SELECT
      SalesOrderID,
         SalesOrderDetailID,
         OrderQty,
         #SALE.ProductID,
         UnitPrice,
         OrderDate,
         TerritoryID,
         CustomerID,
         SalesPersonID,
         coalesce(#SALE.StandardCost,#MaxCost.StandardCost) AS Cost
INTO #SALEFINAL
FROM #SALE
JOIN #MaxCost
ON #SALE.ProductID = #MaxCost.ProductID;

-- show first 100 rows of SALEFINAL

-- Check the missing value of #SALEFINAL
SELECT count(*) - Count(SalesOrderID),
count(*) - Count(SalesOrderDetailID),
count(*) - Count(OrderQty),
count(*) - Count(OrderDate),
count(*) - Count(ProductID),
count(*) - Count(UnitPrice),
count(*) - Count(TerritoryID),
count(*) - Count(CustomerID),
count(*) - Count(SalesOrderID),
count(*) - Count(Cost)
FROM #SALEFINAL;
```



	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)	(No column name)
1	0	0	0	0	0	0	0	0	0

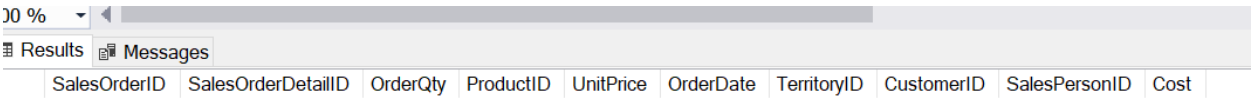
```
-- checking the duplicate data


SELECT SalesOrderID,SalesOrderDetailID
  FROM #SALEFINAL
  GROUP BY SalesOrderID,SalesOrderDetailID
  HAVING COUNT(*) >1

SELECT SalesOrderID,SalesOrderDetailID,OrderQty
  FROM #SALEFINAL
  GROUP BY SalesOrderID,SalesOrderDetailID,OrderQty
  HAVING COUNT(*) >1

SELECT SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate
  FROM #SALEFINAL
  GROUP BY SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate
  HAVING COUNT(*) >1

SELECT SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID
  FROM #SALEFINAL
  GROUP BY SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID
  HAVING COUNT(*) >1

SELECT SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice
  FROM #SALEFINAL
  GROUP BY SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice
  HAVING COUNT(*) >1

SELECT SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice,TerritoryID
  FROM #SALEFINAL
  GROUP BY
SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice,TerritoryID
  HAVING COUNT(*) >1

SELECT
SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice,TerritoryID,Custom
erID
  FROM #SALEFINAL
  GROUP BY
SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice,TerritoryID,Custom
erID
  HAVING COUNT(*) >1
```

```
SELECT
SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice,TerritoryID,Custom
erID,SalesOrderID
  FROM #SALEFINAL
  GROUP BY
SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice,TerritoryID,Custom
erID,SalesOrderID
  HAVING COUNT(*) >1


SELECT
SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice,TerritoryID,Custom
erID,SalesOrderID,Cost
  FROM #SALEFINAL
  GROUP BY
SalesOrderID,SalesOrderDetailID,OrderQty,OrderDate,ProductID,UnitPrice,TerritoryID,Custom
erID,SalesOrderID,Cost
  HAVING COUNT(*) >1
```

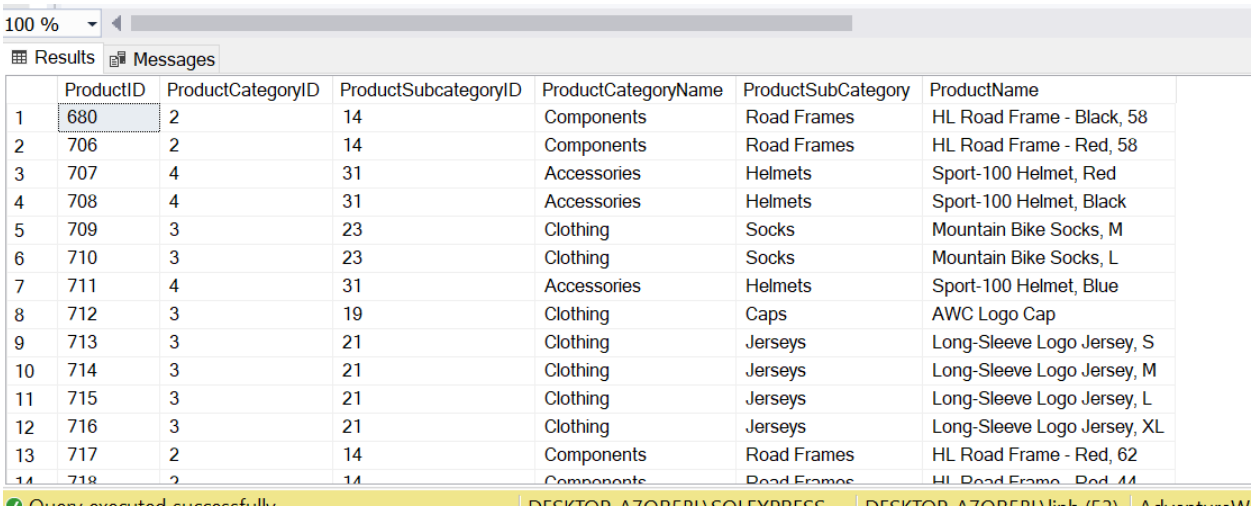



```
-- cheking the invalid data

SELECT *
FROM #SALEFINAL
WHERE ISNUMERIC(SalesOrderID) = 0;

SELECT *
FROM #SALEFINAL
WHERE ISNUMERIC(SalesOrderDetailID) = 0;

SELECT *
FROM #SALEFINAL
WHERE ISNUMERIC(OrderQty) = 0;

SELECT *
FROM #SALEFINAL
WHERE ISDATE(OrderDate) = 0;

SELECT *
FROM #SALEFINAL
WHERE ISNUMERIC(ProductID) = 0;

SELECT *
FROM #SALEFINAL
WHERE ISNUMERIC(UnitPrice) = 0;

SELECT *
```

```
FROM #SALEFINAL
WHERE ISNUMERIC(TerritoryID) = 0;

SELECT *
FROM #SALEFINAL
WHERE ISNUMERIC(CustomerID) = 0;

SELECT *
FROM #SALEFINAL
WHERE ISNUMERIC(SalesOrderID) = 0;

SELECT *
FROM #SALEFINAL
WHERE ISNUMERIC(Cost) = 0;
```




```
-- create Product table
SELECT
       pro.ProductID,
          subca.ProductCategoryID,
          subca.ProductSubcategoryID,
          proca.Name as ProductCategoryName,
          subca.Name AS ProductSubCategory,
          pro.Name AS ProductName

          INTO #PRODUCT
FROM [Production].[ProductCategory] as proca
JOIN [Production].[ProductSubcategory] as subca
ON proca.ProductCategoryID=subca.ProductCategoryID
JOIN [Production].[Product] as pro
ON pro.ProductSubcategoryID = subca.ProductSubcategoryID


SELECT *
  FROM #PRODUCT
```

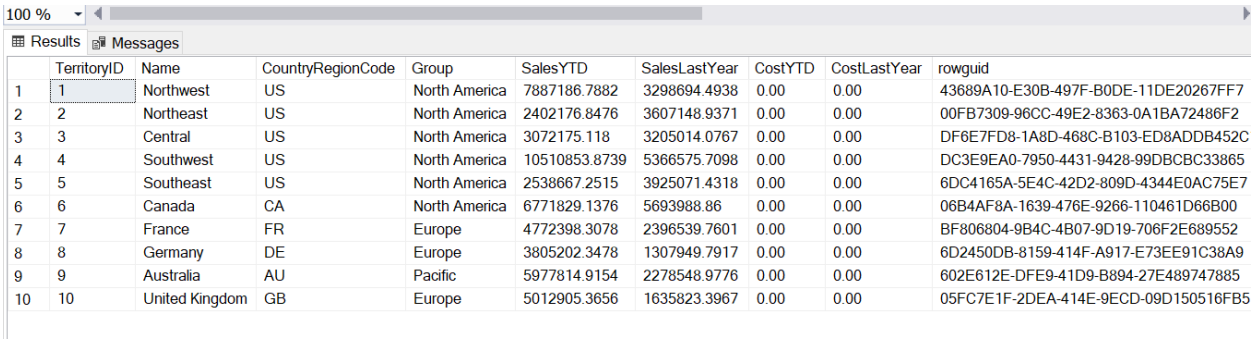


100 %

Results | Messages

	ProductID	ProductCategoryID	ProductSubcategoryID	ProductCategoryName	ProductSubCategory	ProductName
1	680	2	14	Components	Road Frames	HL Road Frame - Black, 58
2	706	2	14	Components	Road Frames	HL Road Frame - Red, 58
3	707	4	31	Accessories	Helmets	Sport-100 Helmet, Red
4	708	4	31	Accessories	Helmets	Sport-100 Helmet, Black
5	709	3	23	Clothing	Socks	Mountain Bike Socks, M
6	710	3	23	Clothing	Socks	Mountain Bike Socks, L
7	711	4	31	Accessories	Helmets	Sport-100 Helmet, Blue
8	712	3	19	Clothing	Caps	AWC Logo Cap
9	713	3	21	Clothing	Jerseys	Long-Sleeve Logo Jersey, S
10	714	3	21	Clothing	Jerseys	Long-Sleeve Logo Jersey, M
11	715	3	21	Clothing	Jerseys	Long-Sleeve Logo Jersey, L
12	716	3	21	Clothing	Jerseys	Long-Sleeve Logo Jersey, XL
13	717	2	14	Components	Road Frames	HL Road Frame - Red, 62
14	718	2	14	Components	Road Frames	HL Road Frame - Red, 44

```sql
SELECT *
  INTO #TERRITORY
  FROM [Sales].[SalesTerritory]

SELECT *
 FROM #TERRITORY
```



100 %

Results | Messages

	TerritoryID	Name	CountryRegionCode	Group	SalesYTD	SalesLastYear	CostYTD	CostLastYear	rowguid
1	1	Northwest	US	North America	7887186.7882	3298694.4938	0.00	0.00	43689A10-E30B-497F-B0DE-11DE20267FF7
2	2	Northeast	US	North America	2402176.8476	3607148.9371	0.00	0.00	00FB7309-96CC-49E2-8363-0A1BA72486F2
3	3	Central	US	North America	3072175.118	3205014.0767	0.00	0.00	DF6E7FD8-1A8D-468C-B103-ED8ADDB452C
4	4	Southwest	US	North America	10510853.8739	5366575.7098	0.00	0.00	DC3E9EA0-7950-4431-9428-99DBCBC33865
5	5	Southeast	US	North America	2538667.2515	3925071.4318	0.00	0.00	6DC4165A-5E4C-42D2-809D-4344E0AC75E7
6	6	Canada	CA	North America	6771829.1376	5693988.86	0.00	0.00	06B4AF8A-1639-476E-9266-110461D66B00
7	7	France	FR	Europe	4772398.3078	2396539.7601	0.00	0.00	BF806804-9B4C-4B07-9D19-706F2E689552
8	8	Germany	DE	Europe	3805202.3478	1307949.7917	0.00	0.00	6D2450DB-8159-414F-A917-E73EE91C38A9
9	9	Australia	AU	Pacific	5977814.9154	2278548.9776	0.00	0.00	602E612E-DFE9-41D9-B894-27E489747885
10	10	United Kingdom	GB	Europe	5012905.3656	1635823.3967	0.00	0.00	05FC7E1F-2DEA-414E-9ECD-09D150516FB5