Project 3: Data Reporting and Analysis with T-SQL

***Project :* Product sales performance analysis using T-SQL**

Submitted by - Clayton Bracken

About the project: Analysis of sales of various products by customers demographics and product categories for Adventure Works Cycles using T-SQL programming on Azure Data Studio.

Aim: Using T-SQL programming to summarize the sales of Adventure Works Cycles with respect to product characteristics, promotion cost and customer demographics.

Objectives: 1. Establish the connection with sql server

1. Generate reports to containing details of the company's customers to support sales campaign

a. Retrieve customer details

Familiarize yourself with the Customer table by writing a Transact-SQL query that retrieves all columns for all customers

```
[1]   1   select * from SalesLT.Customer;
```

(847 rows affected)

Total execution time: 00:00:00.903

	CustomerID	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	CompanyName	SalesPerson	EmailAddress	Phone	P
1	1	0	Mr.	Orlando	N.	Gee	NULL	A Bike Store	adventure-works\pamela0	orlando0@adventure-works.com	245-555-0173	
2	2	0	Mr.	Keith	NULL	Harris	NULL	Progressive Sports	adventure-works\david8	keith0@adventure-works.com	170-555-0127	

b. Retrieve customer name data

Create a list of all customer contact names that includes the title, first name, middle name (if any), last name, and suffix (if any) of all customers.

```
1    SELECT CONCAT(Title,FirstName,MiddleName,LastName,suffix) as Customer_Name from SalesLT.Customer ;
```

(847 rows affected)

Total execution time: 00:00:00.332

	Customer_Name
1	Mr.OrlandoN.Gee
2	Mr.KeithHarris
3	Ms.DonnaF.Carr…
4	Ms.JanetM.Gates
5	Mr.LucyHarring…
6	Ms.RosmarieJ.C…
7	Mr.DominicP.Ga…
8	Ms.KathleenM.G…
9	Ms.KatherineHa…
1…	Mr.JohnnyA.Cap…
1…	Mr.Christopher…
1…	Mr.DavidJ.Liu
1…	Mr.JohnA.Beaver
1…	Ms.JeanP.Handl…
1…	JinghaoLiu
1…	Ms.LindaE.Burn…

C. Retrieve customer names and phone numbers

Each customer has an assigned salesperson. You must write a query to create a call sheet that lists:

• The salesperson

• A column named CustomerName that displays how the customer contact should be greeted (for example, "Mr Smith")

• The customer's phone number.

```
[25]    1    SELECT CONCAT(Title,FirstName,MiddleName,LastName,suffix) as Customer_Name, Phone  from SalesLT.Customer ;
```

(847 rows affected)

Total execution time: 00:00:00.365

	Customer_Name	Phone
1	Mr.OrlandoN.Gee	245-555-0173
2	Mr.KeithHarris	170-555-0127
3	Ms.DonnaF.Carr…	279-555-0130
4	Ms.JanetM.Gates	710-555-0173
5	Mr.LucyHarring…	828-555-0186
6	Ms.RosmarieJ.C…	244-555-0112
7	Mr.DominicP.Ga…	192-555-0173
8	Ms.KathleenM.G…	150-555-0127
9	Ms.KatherineHa…	926-555-0159
1…	Mr.JohnnyA.Cap…	112-555-0191
1…	Mr.Christopher…	1 (11) 500 …

3. Concatenating columns to create reports from same tables

a. Retrieve a list of customer companies

You have been asked to provide a list of all customer companies in the format : - for example, 78: Preferred Bikes.

```
[2]   1   SELECT CustomerID, + ':'+ CompanyName from SalesLT.Customer;
```

(847 rows affected)

Total execution time: 00:00:00.573

	CustomerID	(No column name)
1	1	:A Bike Store
2	2	:Progressive Sports
3	3	:Advanced Bike Com…
4	4	:Modular Cycle Sys…
5	5	:Metropolitan Spor…
6	6	:Aerobic Exercise …
7	7	:Associated Bikes
8	10	:Rural Cycle Empor…
9	11	:Sharp Bikes
1…	12	:Bikes and Motorbi…

b. Retrieve a list of sales order revisions

The SalesLT.SalesOrderHeader table contains records of sales orders. You have been asked to retrieve data for a report that shows:

- The sales order number and revision number in the format () – for example SO71774 (2).
- The order date was converted to ANSI standard format (yyyy.mm.dd – for example 2015.01.31).

```
1   SELECT SalesOrderNumber + ' (' + STR(RevisionNumber, 1) + ')' AS OrderRevision,
2       CONVERT(nvarchar(30), OrderDate, 102) AS OrderDate
3    FROM SalesLT.SalesOrderHeader;
```

(32 rows affected)

Total execution time: 00:00:00.227

	OrderRevision	OrderDate
1	SO71774 (2)	2008.06.01
2	SO71776 (2)	2008.06.01
3	SO71780 (2)	2008.06.01
4	SO71782 (2)	2008.06.01
5	SO71783 (2)	2008.06.01
6	SO71784 (2)	2008.06.01
7	SO71796 (2)	2008.06.01
8	SO71797 (2)	2008.06.01
9	SO71815 (2)	2008.06.01

4. Handling the NULL values in the database

Some records in the database include missing or unknown values that are returned as NULL. You must create some queries that handle these NULL fields appropriately.

a. Retrieve customer contact names with middle names if known

You have been asked to write a query that returns a list of customer names. The list must consist of a single field in the format (for example Keith Harris) if the middle name is unknown, or (for example Jane M. Gates) if a middle name is stored in the database.

```
[2]  1  SELECT FirstName + ' ' + ISNULL(MiddleName + ' ', '')+ LastName AS CustomerName
     2  FROM SalesLT.Customer;
```

(847 rows affected)

Total execution time: 00:00:00.347

	CustomerName
1	Orlando N. Gee
2	Keith Harris
3	Donna F. Carr…
4	Janet M. Gates
5	Lucy Harringt…
6	Rosmarie J. C…
7	Dominic P. Ga…
8	Kathleen M. G…
9	Katherine Har…

b. Retrieve primary contact details

Customers may provide Adventure Works with an email address, a phone number, or both. If an email address is available, then it should be used as the primary contact method; if not, then the phone number should be used. You must write a query that returns a list of customer IDs in one column, and a second column named PrimaryContact that contains the email address if known, and otherwise the phone number.

[24]
```
1  SELECT CustomerID, COALESCE(EmailAddress, Phone) AS PrimaryContact
2  FROM SalesLT.Customer;
```

(847 rows affected)

Total execution time: 00:00:00.325

	CustomerID	PrimaryContact
1	1	orlando0@adventure-works.com
2	2	keith0@adventure-works.com
3	3	donna0@adventure-works.com
4	4	janet1@adventure-works.com
5	5	lucy0@adventure-works.com
6	6	rosmarie0@adventure-works.c…
7	7	dominic0@adventure-works.com
8	10	kathleen0@adventure-works.c…
9	11	katherine0@adventure-works.…
1…	12	johnny0@adventure-works.com
1…	16	christopher1@adventure-work…
1…	18	david20@adventure-works.com

C. Retrieve shipping status

You have been asked to create a query that returns a list of sales order IDs and order dates with a column named ShippingStatus that contains the text "Shipped" for orders with a known ship date, and "Awaiting Shipment" for orders with no ship date.

[23]
```
SELECT SalesOrderID, OrderDate,
    CASE
      WHEN ShipDate IS NULL THEN 'Awaiting Shipment'
      ELSE 'Shipped'
    END AS ShippingStatus
FROM SalesLT.SalesOrderHeader;
```

(32 rows affected)

Total execution time: 00:00:00.215

	SalesOrderID	OrderDate	ShippingStatus
6	71784	2008-06-01 00:00:00.000	Shipped
7	71796	2008-06-01 00:00:00.000	Shipped
8	71797	2008-06-01 00:00:00.000	Shipped
9	71815	2008-06-01 00:00:00.000	Shipped
1…	71816	2008-06-01 00:00:00.000	Shipped
1…	71831	2008-06-01 00:00:00.000	Shipped
1…	71832	2008-06-01 00:00:00.000	Shipped
1…	71845	2008-06-01 00:00:00.000	Shipped
1…	71846	2008-06-01 00:00:00.000	Shipped

5. Querying tables to filter and sort data using

a. Retrieve a list of cities

Initially, you need to produce a list of all of your customers' locations. Write a Transact-SQL query that queries the Address table and retrieves all values for City and StateProvince, removing duplicates.

[22]
```
1  SELECT DISTINCT City, StateProvince
2  FROM SalesLT.Address
```

(272 rows affected)

Total execution time: 00:00:00.410

	City	StateProvince
1	Abingdon	England
2	Albany	Oregon
3	Alhambra	California
4	Alpine	California
5	Arlingt…	Texas
6	Auburn	California
7	Aurora	Ontario
8	Austin	Texas
9	Baldwin…	California
1…	Barrie	Ontario

b. Retrieve the heaviest products

Transportation costs are increasing, and you need to identify the heaviest products. Retrieve the names of the top ten percent of products by weight.

```
1    SELECT TOP 10 PERCENT Name FROM SalesLT.Product ORDER BY Weight DESC;
```

(30 rows affected)

Total execution time: 00:00:00.285

	Name
1	Touring-3000 Blue, 62
2	Touring-3000 Yellow, 62
3	Touring-3000 Blue, 58
4	Touring-3000 Yellow, 58
5	Touring-3000 Blue, 54
6	Touring-3000 Yellow, 54
7	Touring-3000 Yellow, 50
8	Touring-3000 Blue, 50
9	Touring-3000 Blue, 44
1…	Touring-3000 Yellow, 44
1…	Mountain-500 Silver, 52
1…	Mountain-500 Black, 52
1…	Mountain-500 Black, 48
1…	Mountain-500 Silver, 48
1…	Mountain-500 Silver, 44
1…	Mountain-500 Black, 44
1…	Touring-2000 Blue, 60

C. Retrieve the heaviest 100 products not including the heaviest ten

The heaviest ten products are transported by a specialist carrier; therefore, you need to modify the previous query to list the heaviest 100 products not including the heaviest ten.

[20]
```
1  SELECT Name FROM SalesLT.Product ORDER BY Weight DESC
2  OFFSET 10 ROWS FETCH NEXT 100 ROWS ONLY;
```

(100 rows affected)

Total execution time: 00:00:00.286

	Name
1	Mountain-500 Silver, 52
2	Mountain-500 Black, 52
3	Mountain-500 Black, 48
4	Mountain-500 Silver, 48
5	Mountain-500 Silver, 44
6	Mountain-500 Black, 44
7	Touring-2000 Blue, 60
8	Mountain-500 Black, 42
9	Mountain-500 Silver, 42
1…	Touring-2000 Blue, 54
1…	Touring-2000 Blue, 50
1…	Mountain-400-W Silver,…
1…	Mountain-500 Silver, 40
1…	Mountain-500 Black, 40

d. Retrieve product details for product model 1

Initially, you need to find the names, colors, and sizes of the products with a product model ID 1.

[19]
```
1   SELECT Name, Color, Size
2   FROM SalesLT.Product
3   WHERE ProductModelID = 1;
```

(3 rows affected)

Total execution time: 00:00:04.259

	Name	Color	Size
1	Classic Vest, S	Blue	S
2	Classic Vest, M	Blue	M
3	Classic Vest, L	Blue	L

e. Filter products by color and size

Retrieve the product number and name of the products that have a color of 'black', 'red', or 'white' and a size of 'S' or 'M'.

[18]
```
1   SELECT ProductNumber, Name
2   FROM SalesLT.Product
3   WHERE Color IN ('Black','Red','White') and Size IN ('S','M');
```

(12 rows affected)

Total execution time: 00:00:00.257

	ProductNumber	Name
1	SO-B909-M	Mountain Bike Socks, M
2	SH-M897-S	Men's Sports Shorts, S
3	SH-M897-M	Men's Sports Shorts, M
4	TG-W091-S	Women's Tights, S
5	TG-W091-M	Women's Tights, M
6	GL-H102-S	Half-Finger Gloves, S
7	GL-H102-M	Half-Finger Gloves, M
8	GL-F110-S	Full-Finger Gloves, S
9	GL-F110-M	Full-Finger Gloves, M
1…	SH-W890-S	Women's Mountain Shor…
1…	SH-W890-M	Women's Mountain Shor…
1…	SO-R809-M	Racing Socks, M

f. Filter products by product number

Retrieve the product number, name, and list price of products whose product number begins 'BK-'.

[17]
```
SELECT ProductNumber, Name, ListPrice
FROM SalesLT.Product
WHERE ProductNumber LIKE 'BK-%';
```

(97 rows affected)

Total execution time: 00:00:00.260

	ProductNumber	Name	ListPrice
1	BK-R93R-62	Road-150 Red, 62	3578.2700
2	BK-R93R-44	Road-150 Red, 44	3578.2700
3	BK-R93R-48	Road-150 Red, 48	3578.2700
4	BK-R93R-52	Road-150 Red, 52	3578.2700
5	BK-R93R-56	Road-150 Red, 56	3578.2700
6	BK-R68R-58	Road-450 Red, 58	1457.9900
7	BK-R68R-60	Road-450 Red, 60	1457.9900
8	BK-R68R-44	Road-450 Red, 44	1457.9900
9	BK-R68R-48	Road-450 Red, 48	1457.9900
1…	BK-R68R-52	Road-450 Red, 52	1457.9900
1…	BK-R50R-58	Road-650 Red, 58	782.9900
1…	BK-R50R-60	Road-650 Red, 60	782.9900
1…	BK-R50R-62	Road-650 Red, 62	782.9900
1…	BK-R50R-44	Road-650 Red, 44	782.9900
1	BK-R50R-48	Road-650 Red, 48	782.9900

6. Querying tables to join multiple tables and generate reports

a. Retrieve customer orders to generate invoice reports

As an initial step towards generating the invoice report, write a query that returns the company name from the SalesLT.Customer table, and the sales order ID and total due from the SalesLT.SalesOrderHeader table.

[15]
```
1  SELECT c.CompanyName, oh.SalesOrderID, oh.TotalDue
2  FROM SalesLT.Customer AS c
3  JOIN SalesLT.SalesOrderHeader AS oh
4  ON oh.CustomerID = c.CustomerID;
```

(32 rows affected)

Total execution time: 00:00:00.495

	CompanyName	SalesOrderID	TotalDue
1	Professional Sales and Servi…	71782	43962.7901
2	Remarkable Bike Store	71935	7330.8972
3	Bulk Discount Store	71938	98138.2131
4	Coalition Bike Company	71899	2669.3183
5	Futuristic Bikes	71895	272.6468
6	Channel Outlet	71885	608.1766
7	Aerobic Exercise Company	71915	2361.6403
8	Vigorous Sports Store	71867	1170.5376
9	Thrilling Bike Tours	71858	15275.1977
1…	Extreme Riding Supplies	71796	63686.2708
1…	Action Bicycle Specialists	71784	119960.82…

b. Retrieve customer orders with addresses

Extend your customer orders query to include the Main Office address for each customer, including the full street address, city, state or province, postal code, and country or region

[16]
```
SELECT c.CompanyName, a.AddressLine1, ISNULL(a.AddressLine2, '') AS AddressLine2,
a.City, a.StateProvince, a.PostalCode, a.CountryRegion, oh.SalesOrderID, oh.TotalDue
FROM SalesLT.Customer AS c
JOIN SalesLT.SalesOrderHeader AS oh
ON oh.CustomerID = c.CustomerID
JOIN SalesLT.CustomerAddress AS ca
ON c.CustomerID = ca.CustomerID AND AddressType = 'Main Office'
JOIN SalesLT.Address AS a
ON ca.AddressID = a.AddressID;
```

(32 rows affected)

Total execution time: 00:00:00.289

	CompanyName	AddressLine1	AddressLine2	City	StateProvince	PostalCode	CountryRegion	SalesOrderID	TotalDue
1	Good Toys	99700 Bell Road		Auburn	California	95603	United States	71774	972.7850
2	West Side Mart	251 The Metro Center		Wokingham	England	RG41 1QW	United Kingdom	71776	87.0851
3	Nearby Cycle …	Burgess Hill	Edward Way	West Sus…	England	RH15 9UD	United Kingdom	71780	42452.6…
4	Professional …	57251 Serene Blvd		Van Nuys	California	91411	United States	71782	43962.7…
5	Eastside Depa…	9992 Whipple Rd		Union Ci…	California	94587	United States	71783	92663.5…
6	Action Bicycl…	Warrington Ldc Unit…		Woolston	England	WA1 4SY	United Kingdom	71784	119960.…
7	Extreme Ridin…	Riverside		Sherman …	California	91403	United States	71796	63686.2…
8	Riding Cycles	Galashiels		Liverpool	England	L4 4HB	United Kingdom	71797	86222.8…
9	Thrifty Parts…	Oxnard Outlet		Oxnard	California	93030	United States	71815	1261.44…
1…	Engineered Bi…	123 Camelia Avenue		Oxnard	California	93030	United States	71816	3754.97…

C. Retrieve a list of all customers and their orders

The sales manager wants a list of all customer companies and their contacts (first name and last name), showing the sales order ID and total due for each order they have placed. Customers who have not placed any orders should be included at the bottom of the list with NULL values for the order ID and total due.

[14]
```
1  SELECT c.CompanyName, c.FirstName, c.LastName, oh.SalesOrderID, oh.TotalDue
2  FROM SalesLT.Customer AS c
3  LEFT JOIN SalesLT.SalesOrderHeader AS oh
4  ON c.CustomerID = oh.CustomerID
5  ORDER BY oh.SalesOrderID DESC;
```

(847 rows affected)

Total execution time: 00:00:00.336

	CompanyName	FirstName	LastName	SalesOrderID	TotalDue
1	Central Bicycle Specialists	Janeth	Esteves	71946	43.0437
2	Bulk Discount Store	Christopher	Beck	71938	98138.2131
3	Metropolitan Bicycle Supply	Krishna	Sunkam…	71936	108597.95…
4	Remarkable Bike Store	Cory	Booth	71935	7330.8972
5	The Bicycle Accessories Co…	Guy	Gilbert	71923	117.7276
6	Discount Tours	Melissa	Marple	71920	3293.7761
7	Essential Bike Works	Linda	Mitche…	71917	45.1995
8	Aerobic Exercise Company	Rosmarie	Carroll	71915	2361.6403
9	Many Bikes Store	Jeffrey	Kurtz	71902	81834.9826
1…	Coalition Bike Company	Donald	Blanton	71899	2669.3183
1…	Instruments and Parts Comp…	Rebecca	Laszlo	71898	70698.9922
1…	Paints and Solvents Company	Joyce	Jarvis	71897	14017.9083
1…	Futuristic Bikes	Walter	Brian	71895	272.6468

d. Retrieve a list of customers with no address

A sales employee has noticed that AdventureWorks does not have address information for all customers. You must write a query that returns a list of customer IDs, company names, contact names (first name and last name), and phone numbers for customers with no address stored in the database.

[13]
```
1  SELECT c.CompanyName, c.FirstName, c.LastName, c.Phone
2  FROM SalesLT.Customer AS c
3  LEFT JOIN SalesLT.CustomerAddress AS ca
4  ON c.CustomerID = ca.CustomerID
5  WHERE ca.AddressID IS NULL;
```

(440 rows affected)

Total execution time: 00:00:00.310

	CompanyName	FirstName	LastName	Phone
1	A Bike Store	Orlando	Gee	245-555-0173
2	Progressive Sports	Keith	Harris	170-555-0127
3	Advanced Bike Com…	Donna	Carre…	279-555-0130
4	Modular Cycle Sys…	Janet	Gates	710-555-0173
5	Metropolitan Spor…	Lucy	Harri…	828-555-0186
6	Aerobic Exercise …	Rosmar…	Carro…	244-555-0112
7	Associated Bikes	Dominic	Gash	192-555-0173
8	Rural Cycle Empor…	Kathle…	Garza	150-555-0127
9	Sharp Bikes	Kather…	Hardi…	926-555-0159
1…	Bikes and Motorbi…	Johnny	Caprio	112-555-0191
1…	Bulk Discount Sto…	Christ…	Beck	1 (11) 500 …

e. Retrieve a list of customers and products without orders

Some customers have never placed orders, and some products have never been ordered. Create a query that returns a column of customer IDs for customers who have never placed an order, and a column of product IDs for products that have never been ordered. Each row with a customer ID should have a NULL product ID (because the customer has never ordered a product) and each row with a product ID should have a NULL customer ID (because the product has never been ordered by a customer).

[12]
```sql
SELECT c.CustomerID, p.ProductID
FROM SalesLT.Customer AS c
FULL JOIN SalesLT.SalesOrderHeader AS oh
ON c.CustomerID = oh.CustomerID
FULL JOIN SalesLT.SalesOrderDetail AS od
ON od.SalesOrderID = oh.SalesOrderID
FULL JOIN SalesLT.Product AS p
ON p.ProductID = od.ProductID
WHERE oh.SalesOrderID IS NULL
ORDER BY ProductID, CustomerID;
```

(968 rows affected)

Total execution time: 00:00:00.417

	CustomerID	ProductID
1	1	NULL
2	2	NULL
3	3	NULL
4	4	NULL
5	5	NULL
6	6	NULL
7	7	NULL
8	10	NULL
9	11	NULL

7. Working with conditions, aggregation and sub-queries in TSQL

Adventure Works products each have a standard cost price that indicates the cost of manufacturing the product, and a list price that indicates the recommended selling price for the product. This data is stored in the SalesLT.Product table. Whenever a product is ordered, the actual unit price at which it was sold is also recorded in the SalesLT.SalesOrderDetail table. You must use subqueries to compare the cost and list prices for each product with the unit prices charged in each sale.

1. Retrieve products whose list price is higher than the average unit price

Retrieve the product ID, name, and list price for each product where the list price is higher than the average unit price for all products that have been sold.

[11]
```
1   SELECT ProductID, Name, ListPrice from SalesLT.Product
2   WHERE ListPrice >
3   (SELECT AVG(UnitPrice) FROM SalesLT.SalesOrderDetail)
4   ORDER BY ProductID;
```

(137 rows affected)

Total execution time: 00:00:00.959

	ProductID	Name	ListPrice
1	680	HL Road Frame - Black, 58	1431.5000
2	706	HL Road Frame - Red, 58	1431.5000
3	717	HL Road Frame - Red, 62	1431.5000
4	718	HL Road Frame - Red, 44	1431.5000
5	719	HL Road Frame - Red, 48	1431.5000
6	720	HL Road Frame - Red, 52	1431.5000

1. Retrieve Products with a list price of $100 or more that have been sold for less than $100

Retrieve the product ID, name, and list price for each product where the list price is $100 or more, and the product has been sold for less than $100.

[9]
```
SELECT ProductID, Name, ListPrice FROM SalesLT.Product
WHERE ProductID IN
(SELECT ProductID from SalesLT.SalesOrderDetail
 WHERE UnitPrice < 100.00)
AND ListPrice >= 100.00
ORDER BY ProductID;
```

(7 rows affected)

Total execution time: 00:00:00.271

	ProductID	Name	ListPrice
1	810	HL Mountain Handlebars	120.2700
2	813	HL Road Handlebars	120.2700
3	876	Hitch Rack - 4-Bike	120.0000
4	894	Rear Derailleur	121.4600
5	907	Rear Brakes	106.5000
6	948	Front Brakes	106.5000
7	996	HL Bottom Bracket	121.4900

1. Retrieve the cost, list price, and average selling price for each product

Retrieve the product ID, name, cost, and list price for each product along with the average unit price for which that product has been sold.

[8]
```
SELECT ProductID, Name, StandardCost, ListPrice,
    (SELECT AVG(UnitPrice)
     FROM SalesLT.SalesOrderDetail AS SOD
     WHERE P.ProductID = SOD.ProductID) AS AvgSellingPrice
FROM SalesLT.Product AS P
ORDER BY P.ProductID;
```

(295 rows affected)

Total execution time: 00:00:00.284

	ProductID	Name	StandardCost	ListPrice	AvgSellingPrice
1	680	HL Road Frame - Black, 58	1059.3100	1431.5000	NULL
2	706	HL Road Frame - Red, 58	1059.3100	1431.5000	NULL
3	707	Sport-100 Helmet, Red	13.0863	34.9900	20.9940
4	708	Sport-100 Helmet, Black	13.0863	34.9900	20.6441
5	709	Mountain Bike Socks, M	3.3963	9.5000	NULL
6	710	Mountain Bike Socks, L	3.3963	9.5000	NULL
7	711	Sport-100 Helmet, Blue	13.0863	34.9900	20.7440
8	712	AWC Logo Cap	6.9223	8.9900	5.3740
9	713	Long-Sleeve Logo Jersey,…	38.4923	49.9900	NULL
1…	714	Long-Sleeve Logo Jersey,…	38.4923	49.9900	29.9940
1…	715	Long-Sleeve Logo Jersey,…	38.4923	49.9900	29.7440

1. 1. Retrieve products that have an average selling price that is lower than the cost

   Filter your previous query to include only products where the cost price is higher than the average selling price

[10]
```
SELECT ProductID, Name, StandardCost, ListPrice,
(SELECT AVG(UnitPrice)
 FROM SalesLT.SalesOrderDetail AS SOD
 WHERE P.ProductID = SOD.ProductID) AS AvgSellingPrice
FROM SalesLT.Product AS P
WHERE StandardCost >
(SELECT AVG(UnitPrice)
 FROM SalesLT.SalesOrderDetail AS SOD
 WHERE P.ProductID = SOD.ProductID)
ORDER BY P.ProductID;
```

(60 rows affected)

Total execution time: 00:00:00.301

	ProductID	Name	StandardCost	ListPrice	AvgSellingPrice
1	712	AWC Logo Cap	6.9223	8.9900	5.3740
2	714	Long-Sleeve Logo Jersey, M	38.4923	49.9900	29.9940
3	715	Long-Sleeve Logo Jersey, L	38.4923	49.9900	29.7440
4	716	Long-Sleeve Logo Jersey, …	38.4923	49.9900	29.9940
5	717	HL Road Frame - Red, 62	868.6342	1431.5…	858.9000
6	718	HL Road Frame - Red, 44	868.6342	1431.5…	858.9000
7	722	LL Road Frame - Black, 58	204.6251	337.22…	202.3320
8	738	LL Road Frame - Black, 52	204.6251	337.22…	202.3320
9	792	Road-250 Red, 58	1554.9479	2443.3…	1466.0100