#VIPatAINEAI

Project 3: Data Reporting and Analysis with T-SQL

Submitted By: Umadevi Balasubramanian

Data Science Intern – September 2021

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

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:

Question1: Establish connection with SQL servers.

Question 2: Generate reports to containing details of the company's customers to support sales campaign.

2. a) Retrieve customer details.

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

Ans:
```
SELECT * FROM [SalesLT].[Customer];
```

Output – 2.a

Results Messages

	CustomerID	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	CompanyName	SalesPerson	EmailAddress
1	1	0	Mr.	Orlando	N.	Gee	NULL	A Bike Store	adventure-works\pamela0	orlando0@adventure-works.co
2	2	0	Mr.	Keith	NULL	Harris	NULL	Progressive Sports	adventure-works\david8	keith0@adventure-works.com
3	3	0	Ms.	Donna	F.	Carreras	NULL	Advanced Bike Components	adventure-works\jillian0	donna0@adventure-works.com
4	4	0	Ms.	Janet	M.	Gates	NULL	Modular Cycle Systems	adventure-works\jillian0	janet1@adventure-works.com
5	5	0	Mr.	Lucy	NULL	Harrington	NULL	Metropolitan Sports Supply	adventure-works\shu0	lucy0@adventure-works.com
6	6	0	Ms.	Rosmarie	J.	Carroll	NULL	Aerobic Exercise Company	adventure-works\linda3	rosmarie0@adventure-works.c
7	7	0	Mr.	Dominic	P.	Gash	NULL	Associated Bikes	adventure-works\shu0	dominic0@adventure-works.co
8	10	0	Ms.	Kathleen	M.	Garza	NULL	Rural Cycle Emporium	adventure-works\josé1	kathleen0@adventure-works.c
9	11	0	Ms.	Katherine	NULL	Harding	NULL	Sharp Bikes	adventure-works\josé1	katherine0@adventure-works.
1…	12	0	Mr.	Johnny	A.	Caprio	Jr.	Bikes and Motorbikes	adventure-works\garrett1	johnny0@adventure-works.com
1…	16	0	Mr.	Christopher	R.	Beck	Jr.	Bulk Discount Store	adventure-works\jae0	christopher1@adventure-work
1…	18	0	Mr.	David	J.	Liu	NULL	Catalog Store	adventure-works\michael9	david20@adventure-works.com
1…	19	0	Mr.	John	A.	Beaver	NULL	Center Cycle Shop	adventure-works\pamela0	john8@adventure-works.com
1…	20	0	Ms.	Jean	P.	Handley	NULL	Central Discount Store	adventure-works\david8	jean1@adventure-works.com
1…	21	0	NULL	Jinghao	NULL	Liu	NULL	Chic Department Stores	adventure-works\jillian0	jinghao1@adventure-works.co
1…	22	0	Ms.	Linda	E.	Burnett	NULL	Travel Systems	adventure-works\jillian0	linda4@adventure-works.com
1…	23	0	Mr.	Kerim	NULL	Hanif	NULL	Bike World	adventure-works\shu0	kerim0@adventure-works.com
1…	24	0	Mr.	Kevin	NULL	Liu	NULL	Eastside Department Store	adventure-works\linda3	kevin5@adventure-works.com

EmailAddress	Phone	PasswordHash	PasswordSalt	rowguid	ModifiedDate
gregory1@adventure-works.com	684-555-0134	syI1UO2qeBH9g2tg2nu3DTejZc7O…	FAw6ojc=	67ca966e-b20f-4230-b81c-42f0…	2006-09-01 00:00:00.000
michael25@adventure-works.com	918-555-0141	/upSM1yXqGCA1vtYL9IB/NcKFrE3…	uXetzTc=	ca0d950a-a3fe-41c6-9b1a-125a…	2005-09-01 00:00:00.000
margaret2@adventure-works.com	265-555-0143	TUk6pzHyHoA6CadqZKdfwfCKtp3f…	osDZuYo=	ac07da4a-9fb0-4832-8219-5b6d…	2005-08-01 00:00:00.000
kara0@adventure-works.com	680-555-0160	grGra0LtVSrZMZySvhV+ggFwVRSD…	kc4E3DA=	a588ee77-50ee-445f-989b-e8dd…	2006-08-01 00:00:00.000
nieves0@adventure-works.com	371-555-0184	dG3/Y2fWXOAgE37z4Vj0K7ejAxDb…	/PSiwRg=	71f5e3bb-78bd-426f-b75a-eeb7…	2006-07-01 00:00:00.000
gary6@adventure-works.com	112-555-0176	WH8bTbYkazE9T0ktiZYIAYDdJXsv…	Zkv5OCw=	4460c7aa-fe06-4d56-98cb-6de6…	2006-09-01 00:00:00.000
ranjit1@adventure-works.com	810-555-0160	VTpXtJAiSxeJraHh5mkwwAsGYMll…	FNe76Vk=	d414d91f-8abe-47de-a576-ddd8…	2005-08-01 00:00:00.000
patricia2@adventure-works.com	490-555-0132	v0bxLRLRKYxwN8fiORyJkpXaYZpC…	WTZFkow=	b2e90958-4a9b-40f9-a8d8-0916…	2006-08-01 00:00:00.000
raja0@adventure-works.com	1 (11) 500 555-0195	lx5a4+AFGzH6mzjz6hpiR9scxbYG…	hSwmBWk=	84975c44-dc82-49d3-bfc4-8f31…	2006-09-01 00:00:00.000
dora0@adventure-works.com	155-555-0140	RgzxIE8hSY/z6iJnmDYMZWdJTQBR…	jYD20wc=	e4cf8fd5-30a4-4b8e-8fd8-4703…	2006-08-01 00:00:00.000
wanda0@adventure-works.com	433-555-0168	hNQpZV8787KyeDWfllnZWzY+18DF…	HDCU1Bk=	ec409609-d25d-41b8-9d15-a1aa…	2007-07-01 00:00:00.000
robert13@adventure-works.com	560-555-0171	UWGC2U8F7AUNA2FuiT4agrBoxAFs…	iES3IZA=	6f08e2fb-1cd3-4f6e-a2e6-3856…	2005-08-01 00:00:00.000
caroline0@adventure-works.com	695-555-0158	U1/CrPqSzwLTtwgBehfpI17f1LHS…	QhHP+y8=	2495b4eb-fe8b-459e-a1b6-dba2…	2006-09-01 00:00:00.000

Ln 25, Col 37 Spaces: 4 UTF-8 CRLF SQL Choose SQL Language 847 rows 00:00:07 sqlservercentralpublic.database.windows.net : AdventureWorks

There are **847 rows** of data are there in Customer table. This query displays the various details of all customers. That is all data from all columns is displayed here.

2. 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.

Ans:
```
SELECT FirstName FROM [SalesLT].[Customer];
```

Output – 2.b

Results Messages

	FirstName
1	Orlando
2	Keith
3	Donna
4	Janet
5	Lucy
6	Rosmarie
7	Dominic
8	Kathleen
9	Katherine
1…	Johnny
1…	Christopher
1…	David

The FirstName of **all the customers** is returned here.

2. 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.

Ans:

```
SELECT Title + ' ' + FirstName as CustomerName, SalesPerson, Phone FROM [Sale
sLT].[Customer] ;
```

Output – 2.C

Results Messages

	CustomerName	SalesPerson	Phone
1	Mr. Orlando	adventure-works\pamela0	245-555-0173
2	Mr. Keith	adventure-works\david8	170-555-0127
3	Ms. Donna	adventure-works\jillian0	279-555-0130
4	Ms. Janet	adventure-works\jillian0	710-555-0173
5	Mr. Lucy	adventure-works\shu0	828-555-0186
6	Ms. Rosmarie	adventure-works\linda3	244-555-0112
7	Mr. Dominic	adventure-works\shu0	192-555-0173
8	Ms. Kathleen	adventure-works\josé1	150-555-0127
9	Ms. Katherine	adventure-works\josé1	926-555-0159
10	Mr. Johnny	adventure-works\garrett1	112-555-0191
11	Mr. Christopher	adventure-works\jae0	1 (11) 500 …
12	Mr. David	adventure-works\michael9	440-555-0132
13	Mr. John	adventure-works\pamela0	521-555-0195
14	Ms. Jean	adventure-works\david8	582-555-0113

The Title, FirstName, SalesPerson and the Phone details of all **847 customers** will be returned.

Question 3: Concatenating columns to create reports from same tables.

3. 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.

Ans:

```
SELECT CAST(CustomerID AS varchar) + ': ' + CompanyName AS CustomerCompany FR
OM [SalesLT].[Customer] ;
```

Output – 3.a

Results Messages

	CustomerCompany
1	1: A Bike Store
2	2: Progressive Sports
3	3: Advanced Bike Components
4	4: Modular Cycle Systems
5	5: Metropolitan Sports Supply
6	6: Aerobic Exercise Company
7	7: Associated Bikes
8	10: Rural Cycle Emporium
9	11: Sharp Bikes
1…	12: Bikes and Motorbikes

The output will be a list of all customer companies along with customer ID for all **847 customers**.

3. 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).

Ans:

```
SELECT SalesOrderNumber + '(' + STR(RevisionNumber, 1) + ')' AS OrderRevision
, CONVERT(nvarchar(30), OrderDate, 102) AS OrderDate FROM [SalesLT].[SalesOrd
erHeader] ;
```

Output – 3.b

Results Messages

	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
10	SO71816(2)	2008.06.01

This output contains **32 records** which shows the Sales Order Number and Revision Number as OrderRevision column and OrderDate from SalesOrderHeader table.

Question 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.

4. 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.

Ans:

```
SELECT FirstName + ' ' + MiddleName + ' ' + LastName AS CustomerName FROM [Sa
lesLT].[Customer] WHERE MiddleName IS NOT NULL;
```

Output – 4.a

Results Messages

	CustomerName
1	Orlando N. Gee
2	Donna F. Carreras
3	Janet M. Gates
4	Rosmarie J. Carroll
5	Dominic P. Gash
6	Kathleen M. Garza
7	Johnny A. Caprio
8	Christopher R. Beck
9	David J. Liu
10	John A. Beaver

The output of the above query will be the CustomerName column which contains the full name of customers who have middle names. If the customer is not having a middle name, that record will not be displayed. So there are **504 customers** with middle names.

4. 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.

Ans:

```
SELECT CustomerID, COALESCE(EmailAddress, Phone) AS PrimaryContact  FROM [Sal
esLT].[Customer] ;
```

Output – 4.b

Results Messages

	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.com
7	7	dominic0@adventure-works.com
8	10	kathleen0@adventure-works.com
9	11	katherine0@adventure-works.c…
10	12	johnny0@adventure-works.com

The output contains the PrimaryContact detail of all **847 customers** with their CustomerID detail.

4.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.

Ans:

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

Output – 4.c

Results Messages

	SalesOrderID	OrderDate	ShippingStatus
1	71774	2008-06-01 00:00:00.000	Shipped
2	71776	2008-06-01 00:00:00.000	Shipped
3	71780	2008-06-01 00:00:00.000	Shipped
4	71782	2008-06-01 00:00:00.000	Shipped
5	71783	2008-06-01 00:00:00.000	Shipped
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
10	71816	2008-06-01 00:00:00.000	Shipped
11	71831	2008-06-01 00:00:00.000	Shipped
12	71832	2008-06-01 00:00:00.000	Shipped
13	71845	2008-06-01 00:00:00.000	Shipped
14	71846	2008-06-01 00:00:00.000	Shipped

This query retrieves the orders with the known ShipDate. There are **32 records** in SalesOrderHeader table whose shipping date is known.

Question 5: Querying Tables to filter and sort data.

5. 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.

Ans:
```
SELECT DISTINCT City, StateProvince FROM [SalesLT].[Address] ;
```

Output – 5.a

Results Messages

	City	StateProvince
1	Abingdon	England
2	Albany	Oregon
3	Alhambra	California
4	Alpine	California
5	Arlington	Texas
6	Auburn	California
7	Aurora	Ontario
8	Austin	Texas
9	Baldwin Park	California
10	Barrie	Ontario
11	Barstow	California

There are **272 distinct City and StateProvince** in Address table, which is displayed here**.**

5. 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.

Ans: `SELECT TOP 10 PERCENT Name FROM [SalesLT].[Product] ORDER BY weight DESC;`

Output – 5.b

Results Messages

	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
10	Touring-3000 Yellow, 44
11	Mountain-500 Silver, 52

Top 10% **(30 records)** of the heaviest products are shown here.

5. 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. (Hint: Use OFFSET and FETCH NEXT)

Ans:
```
SELECT Name FROM [SalesLT].[Product] ORDER BY weight DESC
OFFSET 3 ROWS FETCH NEXT 100 ROWS ONLY;
```

Output – 5.c

Results Messages

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

100 rows of data which shows the heaviest 100 products excluding the top 10.

5. 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.

Ans :
```
SELECT Name, Color, Size FROM [SalesLT].[Product] WHERE ProductModelID = 1 ;
```

Output – 5.d

Results Messages

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

There are **3 records** with ProductModelID 1 whose Name, Color and Size details were displayed.

5. 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'.

Ans:
```
SELECT ProductNumber, Name FROM [SalesLT].[Product] WHERE Color IN ('Black',
'Red', 'White ') AND Size IN ('S', 'M');
```

Output – 5.e

Results Messages

	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
10	SH-W890-S	Women's Mountain Shorts, S
11	SH-W890-M	Women's Mountain Shorts, M

There are **12 records** with Color is either black or red or white and with Size as small/medium.

5. f) Filter products by product number.

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

Ans:

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

Output – 5.f

Results Messages

	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
10	BK-R68R-52	Road-450 Red, 52	1457.9900
11	BK-R50R-58	Road-650 Red, 58	782.9900
12	BK-R50R-60	Road-650 Red, 60	782.9900

97 records are there in the database whose ProductNumber starts with 'BK-'.

Question 6: Querying Tables to join multiple tables and generate reports.

6. 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.

Ans:

```
SELECT c.CompanyName, soh.SalesOrderID, soh.TotalDue FROM [SalesLT].[Customer
] AS c
JOIN [SalesLT].[SalesOrderHeader] AS soh
ON soh.CustomerID = c.CustomerID;
```

Output – 6.a

Results Messages

	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
10	Extreme Riding Supplies	71796	63686.2708
11	Action Bicycle Specialists	71784	119960.8240
12	Central Bicycle Specialists	71946	43.0437

The CompanyName, SalesOrderID and TotalDue for all the **32 records** are displayed from the SalesOrderHeader table.

6. 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.

Ans:

```
SELECT c.CompanyName, a.AddressLine1, ISNULL(a.AddressLine2, '') AS AddressLine2,
a.City, a.StateProvince, a.PostalCode,a.CountryRegion, soh.SalesOrderID, soh.TotalDue
FROM SalesLT.Customer AS c
JOIN SalesLT.SalesOrderHeader AS soh
ON soh.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;
```

Output – 6.b

Results Messages

	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 Shop	Burgess Hill	Edward Way	West Sussex	England	RH15 9UD	United Kingdom	71780	42452.6519
4	Professional Sales and Servi…	57251 Serene Blvd		Van Nuys	California	91411	United States	71782	43962.7901
5	Eastside Department Store	9992 Whipple Rd		Union City	California	94587	United States	71783	92663.5609
6	Action Bicycle Specialists	Warrington Ldc Unit 25/2		Woolston	England	WA1 4SY	United Kingdom	71784	119960.8240
7	Extreme Riding Supplies	Riverside		Sherman Oaks	California	91403	United States	71796	63686.2708
8	Riding Cycles	Galashiels		Liverpool	England	L4 4HB	United Kingdom	71797	86222.8072
9	Thrifty Parts and Sales	Oxnard Outlet		Oxnard	California	93030	United States	71815	1261.4440
10	Engineered Bike Systems	123 Camelia Avenue		Oxnard	California	93030	United States	71816	3754.9733
11	Tachometers and Accessories	Wymbush		Milton Keynes	England	MK8 8DF	United Kingdom	71831	2228.0566
12	Closest Bicycle Store	Garamonde Drive, Wymbush	PO Box 4023	Milton Keynes	England	MK8 8ZD	United Kingdom	71832	39531.6085

32 rows of data with the Main Office address which includes the full street address, City, State or Province, Postal code and Country for each customer is shown here.

6. 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.

Ans:

```
SELECT c.CompanyName, c.FirstName, c.LastName, soh.SalesOrderID, soh.TotalDue
FROM SalesLT.Customer AS c
LEFT JOIN SalesLT.SalesOrderHeader AS soh
ON c.CustomerID = soh.CustomerID
ORDER BY soh.SalesOrderID DESC;
```

Output – 6.c

Results Messages

	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	Sunkammurali	71936	108597.9536
4	Remarkable Bike Store	Cory	Booth	71935	7330.8972
5	The Bicycle Accessories Comp…	Guy	Gilbert	71923	117.7276
6	Discount Tours	Melissa	Marple	71920	3293.7761
7	Essential Bike Works	Linda	Mitchell	71917	45.1995
8	Aerobic Exercise Company	Rosmarie	Carroll	71915	2361.6403
9	Many Bikes Store	Jeffrey	Kurtz	71902	81834.9826
10	Coalition Bike Company	Donald	Blanton	71899	2669.3183
11	Instruments and Parts Company	Rebecca	Laszlo	71898	70698.9922
12	Paints and Solvents Company	Joyce	Jarvis	71897	14017.9083

The Order details for all the **847 customers** is displayed here.

6. 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.

Ans:

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

Output – 6.d

	CompanyName	FirstName	LastName	Phone
1	A Bike Store	Orlando	Gee	245-555-0173
2	Progressive Sports	Keith	Harris	170-555-0127
3	Advanced Bike Components	Donna	Carreras	279-555-0130
4	Modular Cycle Systems	Janet	Gates	710-555-0173
5	Metropolitan Sports Supply	Lucy	Harrington	828-555-0186
6	Aerobic Exercise Company	Rosmarie	Carroll	244-555-0112
7	Associated Bikes	Dominic	Gash	192-555-0173
8	Rural Cycle Emporium	Kathleen	Garza	150-555-0127
9	Sharp Bikes	Katherine	Harding	926-555-0159
10	Bikes and Motorbikes	Johnny	Caprio	112-555-0191
11	Bulk Discount Store	Christopher	Beck	1 (11) 500 555-0132
12	Catalog Store	David	Liu	440-555-0132

There are **440 customers** whose address is not stored in the database.

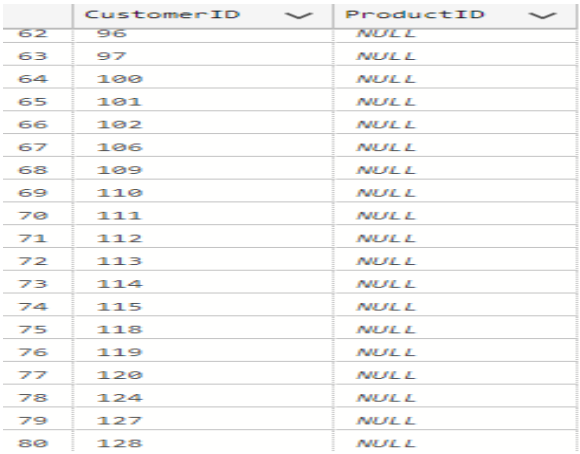
6. e) Retrieve a list of customers and products without orders.

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).

Ans:

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

Output -6.e



	CustomerID	ProductID
62	96	NULL
63	97	NULL
64	100	NULL
65	101	NULL
66	102	NULL
67	106	NULL
68	109	NULL
69	110	NULL
70	111	NULL
71	112	NULL
72	113	NULL
73	114	NULL
74	115	NULL
75	118	NULL
76	119	NULL
77	120	NULL
78	124	NULL
79	127	NULL
80	128	NULL

The customers who have never placed orders and Products which have never been ordered are shown here. They are **totally 968**.

Question 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.

7. a) 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.

Ans:

```
SELECT ProductID, Name, ListPrice FROM [SalesLT].[Product]
WHERE ListPrice > (SELECT AVG (UnitPrice) FROM SalesLT.SalesOrderDetail)
ORDER BY ProductID;
```

Output – 7.a

Results Messages

	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
7	721	HL Road Frame - Red, 56	1431.5000
8	731	ML Road Frame - Red, 44	594.8300
9	732	ML Road Frame - Red, 48	594.8300
10	733	ML Road Frame - Red, 52	594.8300
11	734	ML Road Frame - Red, 58	594.8300

137 products were sold whose ListPrice is higher than the average unit price.

7. b) 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.

Ans:

```
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;
```

Output – 7.b

Results Messages

	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

There are **7 products** whose ListPrice is $100 or more, but have been sold for less than $100.

7. c) 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.

Ans:

```
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;
```

Output – 7.c

Results Messages

	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, S	38.4923	49.9900	*NULL*
10	714	Long-Sleeve Logo Jersey, M	38.4923	49.9900	29.9940
11	715	Long-Sleeve Logo Jersey, L	38.4923	49.9900	29.7440

The various product details such as ProductID, Name, StandardCost, ListPrice & AvgSellingPrice for all the **295 products** are displayed here.

7. d) 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.

Ans:

```
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;
```

Output – 7.d

Results Messages

	ProductID	Name	StandardCost	ListPrice	AvgSellingPrice
3	715	Long-Sleeve Logo Jersey, L	38.4923	49.9900	29.7440
4	716	Long-Sleeve Logo Jersey, XL	38.4923	49.9900	29.9940
5	717	HL Road Frame - Red, 62	868.6342	1431.5000	858.9000
6	718	HL Road Frame - Red, 44	868.6342	1431.5000	858.9000
7	722	LL Road Frame - Black, 58	204.6251	337.2200	202.3320
8	738	LL Road Frame - Black, 52	204.6251	337.2200	202.3320
9	792	Road-250 Red, 58	1554.9479	2443.3500	1466.0100
10	793	Road-250 Black, 44	1554.9479	2443.3500	1466.0100
11	794	Road-250 Black, 48	1554.9479	2443.3500	1466.0100
12	795	Road-250 Black, 52	1554.9479	2443.3500	1466.0100
13	796	Road-250 Black, 58	1554.9479	2443.3500	1466.0100
14	797	Road-550-W Yellow, 38	713.0798	1120.4900	672.2940
15	798	Road-550-W Yellow, 40	713.0798	1120.4900	672.2940

Out of the 295 total products sold, **60 products** have a CostPrice higher than the AvgSellingPrice.