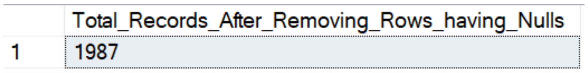
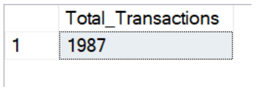
```sql
SELECT TOP (1000) [transactions_id]
      ,[sale_date]
      ,[sale_time]
      ,[customer_id]
      ,[gender]
      ,[age]
      ,[category]
      ,[quantiy]
      ,[price_per_unit]
      ,[cogs]
      ,[total_sale]
  FROM [Retail_Sales].[dbo].[Retail_Sales_Data]

SELECT COUNT(*) Total_Records FROM Retail_Sales_Data

-- Data Cleaning
SELECT * FROM Retail_Sales_Data
WHERE transactions_id IS NULL


SELECT * FROM Retail_Sales_Data
WHERE [sale_date] IS NULL

SELECT * FROM Retail_Sales_Data
WHERE [sale_time] IS NULL
-- We can Check the Nulls by this basic way or by Using OR operator,
Also we can use CASE WHEN method to do so

-- Detecting Nulls using OR operator
SELECT * FROM Retail_Sales_Data
WHERE [customer_id] IS NULL
OR    [gender] IS NULL
OR     [age] IS NULL
OR     [category] IS NULL
OR     [quantiy] IS NULL
OR     [price_per_unit] IS NULL
OR     [cogs] IS NULL
OR     [total_sale] IS NULL

-- Detecting Nulls using Case when function
SELECT *
FROM Retail_Sales_Data
WHERE
    CASE
        WHEN [customer_id] IS NULL THEN 1
        WHEN [gender] IS NULL THEN 1
        WHEN [age] IS NULL THEN 1
        WHEN [category] IS NULL THEN 1
        WHEN [quantiy] IS NULL THEN 1
        WHEN [price_per_unit] IS NULL THEN 1
        WHEN [cogs] IS NULL THEN 1
        WHEN [total_sale] IS NULL THEN 1
        ELSE 0
    END = 1;

-- Now We can delete those Rows having Nulls using Delete function
DELETE FROM Retail_Sales_Data
WHERE
    [customer_id] IS NULL
    OR [gender] IS NULL
    OR [age] IS NULL
    OR [category] IS NULL
    OR [quantiy] IS NULL
    OR [price_per_unit] IS NULL
    OR [cogs] IS NULL
    OR [total_sale] IS NULL;

-- Now the Count of Records is
SELECT COUNT(*) Total_Records_After_Removing_Rows_having_Nulls FROM
Retail_Sales_Data
```



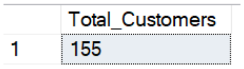
	Total_Records_After_Removing_Rows_having_Nulls
1	1987

```
--Data Exploration
-- How many Transaction we have?
Select COUNT(transactions_id) Total_Transactions from
Retail_Sales_Data
```



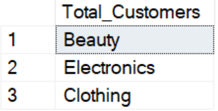
	Total_Transactions
1	1987

```
-- How many Unique Customers we have?
Select COUNT(DISTINCT customer_id) Total_Customers from
Retail_Sales_Data
```



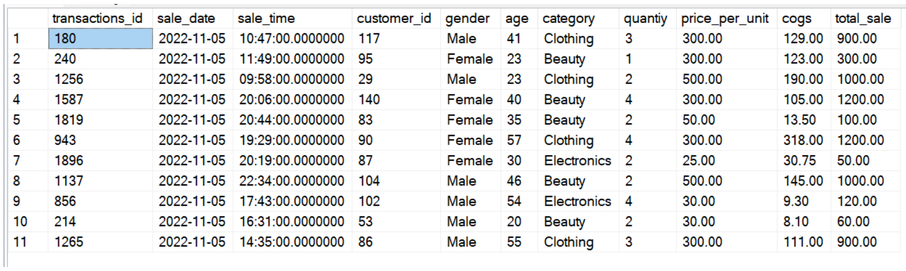
	Total_Customers
1	155

```
-- What are the Unique Categories?
SELECT DISTINCT category Total_Customers from Retail_Sales_Data
```



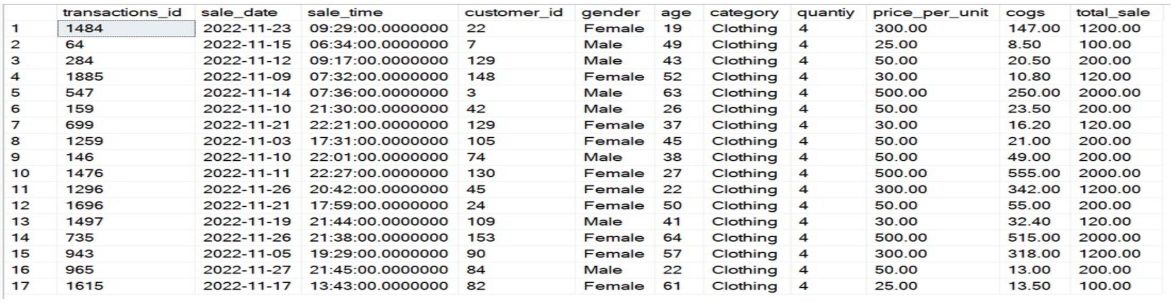
	Total_Customers
1	Beauty
2	Electronics
3	Clothing

```
-- Data analysis and Solution of Business Problems
-- Q.1 Write a SQL query to retrieve all columns for sales made on
'2022-11-05
SELECT * FROM Retail_Sales_Data
WHERE sale_date = '2022-11-05'
```



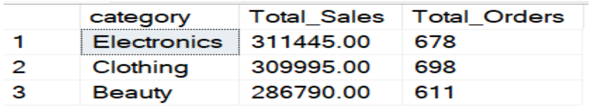
	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
1	180	2022-11-05	10:47:00.0000000	117	Male	41	Clothing	3	300.00	129.00	900.00
2	240	2022-11-05	11:49:00.0000000	95	Female	23	Beauty	1	300.00	123.00	300.00
3	1256	2022-11-05	09:58:00.0000000	29	Male	23	Clothing	2	500.00	190.00	1000.00
4	1587	2022-11-05	20:06:00.0000000	140	Female	40	Beauty	4	300.00	105.00	1200.00
5	1819	2022-11-05	20:44:00.0000000	83	Female	35	Beauty	2	50.00	13.50	100.00
6	943	2022-11-05	19:29:00.0000000	90	Female	57	Clothing	4	300.00	318.00	1200.00
7	1896	2022-11-05	20:19:00.0000000	87	Female	30	Electronics	2	25.00	30.75	50.00
8	1137	2022-11-05	22:34:00.0000000	104	Male	46	Beauty	2	500.00	145.00	1000.00
9	856	2022-11-05	17:43:00.0000000	102	Male	54	Electronics	4	30.00	9.30	120.00
10	214	2022-11-05	16:31:00.0000000	53	Male	20	Beauty	2	30.00	8.10	60.00
11	1265	2022-11-05	14:35:00.0000000	86	Male	55	Clothing	3	300.00	111.00	900.00

```
-- Q.2 Write a SQL query to retrieve all transactions where the
category is 'Clothing'
-- and the quantity sold is more than 10 in the month of Nov-2022

SELECT * FROM Retail_Sales_Data
WHERE category = 'Clothing'  AND FORMAT(sale_date,'MMM yyyy') = 'Nov
2022' and quantiy >= 4
```



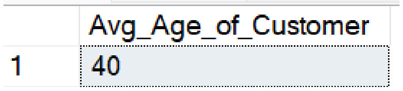
	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
1	1484	2022-11-23	09:29:00.0000000	22	Female	19	Clothing	4	300.00	147.00	1200.00
2	64	2022-11-15	06:34:00.0000000	7	Male	49	Clothing	4	25.00	8.50	100.00
3	284	2022-11-12	09:17:00.0000000	129	Male	43	Clothing	4	50.00	20.50	200.00
4	1885	2022-11-09	07:32:00.0000000	148	Female	52	Clothing	4	30.00	10.80	120.00
5	547	2022-11-14	07:36:00.0000000	3	Male	63	Clothing	4	500.00	250.00	2000.00
6	159	2022-11-10	21:30:00.0000000	42	Male	26	Clothing	4	50.00	23.50	200.00
7	699	2022-11-21	22:21:00.0000000	129	Female	37	Clothing	4	30.00	16.20	120.00
8	1259	2022-11-03	17:31:00.0000000	105	Female	45	Clothing	4	50.00	21.00	200.00
9	146	2022-11-10	22:01:00.0000000	74	Male	38	Clothing	4	50.00	49.00	200.00
10	1476	2022-11-11	22:27:00.0000000	130	Female	27	Clothing	4	500.00	555.00	2000.00
11	1296	2022-11-26	20:42:00.0000000	45	Female	22	Clothing	4	300.00	342.00	1200.00
12	1696	2022-11-21	17:59:00.0000000	24	Female	50	Clothing	4	50.00	55.00	200.00
13	1497	2022-11-19	21:44:00.0000000	109	Male	41	Clothing	4	30.00	32.40	120.00
14	735	2022-11-26	21:38:00.0000000	153	Female	64	Clothing	4	500.00	515.00	2000.00
15	943	2022-11-05	19:29:00.0000000	90	Female	57	Clothing	4	300.00	318.00	1200.00
16	965	2022-11-27	21:45:00.0000000	84	Male	22	Clothing	4	50.00	13.00	200.00
17	1615	2022-11-17	13:43:00.0000000	82	Female	61	Clothing	4	25.00	13.50	100.00

```
-- Q.3 Write a SQL query to calculate the total sales (total_sale) for
each category.

SELECT category,
SUM(total_sale) as Total_Sales,
COUNT(*) Total_Orders
FROM Retail_Sales_Data
GROUP BY category
ORDER BY SUM(total_sale) DESC
```



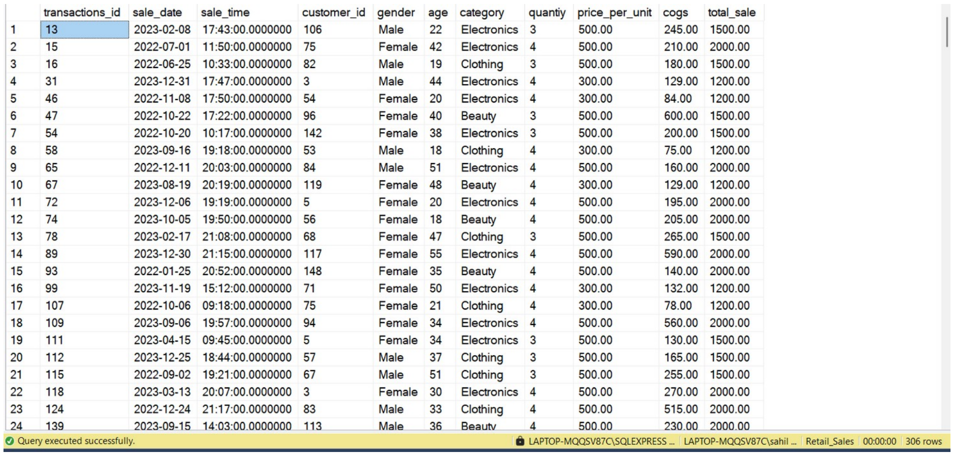
	category	Total_Sales	Total_Orders
1	Electronics	311445.00	678
2	Clothing	309995.00	698
3	Beauty	286790.00	611

```sql
-- Q.4 Write a SQL query to find the average age of customers who
purchased items from the 'Beauty' category.
SELECT
AVG(age) as Avg_Age_of_Customer
FROM Retail_Sales_Data
WHERE category = 'Beauty'
```



	Avg_Age_of_Customer
1	40

```sql
-- Q.5 Write a SQL query to find all transactions where the total_sale
is greater than 1000.
SELECT * FROM Retail_Sales_Data
WHERE total_sale > 1000
ORDER BY transactions_id
```

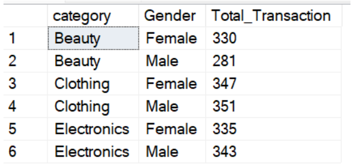


	transactions_id	sale_date	sale_time	customer_id	gender	age	category	quantiy	price_per_unit	cogs	total_sale
1	13	2023-02-08	17:43:00.0000000	106	Male	22	Electronics	3	500.00	245.00	1500.00
2	15	2022-07-01	11:50:00.0000000	75	Female	42	Electronics	4	500.00	210.00	2000.00
3	16	2022-06-25	10:33:00.0000000	82	Male	19	Clothing	3	500.00	180.00	1500.00
4	31	2023-12-31	17:47:00.0000000	3	Male	44	Electronics	4	300.00	129.00	1200.00
5	46	2022-11-08	17:50:00.0000000	54	Female	20	Electronics	4	300.00	84.00	1200.00
6	47	2022-10-22	17:22:00.0000000	96	Female	40	Beauty	3	500.00	600.00	1500.00
7	54	2022-10-20	10:17:00.0000000	142	Female	38	Electronics	3	500.00	200.00	1500.00
8	58	2023-09-16	19:18:00.0000000	53	Male	18	Clothing	4	300.00	75.00	1200.00
9	65	2022-12-11	20:03:00.0000000	84	Male	51	Electronics	4	500.00	160.00	2000.00
10	67	2023-08-19	20:19:00.0000000	119	Female	48	Beauty	4	300.00	129.00	1200.00
11	72	2023-12-06	19:19:00.0000000	5	Female	20	Electronics	4	500.00	195.00	2000.00
12	74	2023-10-05	19:50:00.0000000	56	Female	18	Beauty	4	500.00	205.00	2000.00
13	78	2023-02-17	21:08:00.0000000	68	Female	47	Clothing	3	500.00	265.00	1500.00
14	89	2023-12-30	21:15:00.0000000	117	Female	55	Electronics	4	500.00	590.00	2000.00
15	93	2022-01-25	20:52:00.0000000	148	Female	35	Beauty	4	500.00	140.00	2000.00
16	99	2023-11-19	15:12:00.0000000	71	Female	50	Electronics	4	300.00	132.00	1200.00
17	107	2022-10-06	09:18:00.0000000	75	Female	21	Clothing	4	300.00	78.00	1200.00
18	109	2023-09-06	19:57:00.0000000	94	Female	34	Electronics	4	500.00	560.00	2000.00
19	111	2023-04-15	09:45:00.0000000	5	Female	34	Electronics	3	500.00	130.00	1500.00
20	112	2023-12-25	18:44:00.0000000	57	Male	37	Clothing	3	500.00	165.00	1500.00
21	115	2022-09-02	19:21:00.0000000	67	Male	51	Clothing	3	500.00	255.00	1500.00
22	118	2023-03-13	20:07:00.0000000	3	Female	30	Electronics	4	500.00	270.00	2000.00
23	124	2022-12-24	21:17:00.0000000	83	Male	33	Clothing	4	500.00	515.00	2000.00
24	139	2023-09-15	14:03:00.0000000	113	Male	36	Beauty	4	500.00	230.00	2000.00

✓ Query executed successfully.

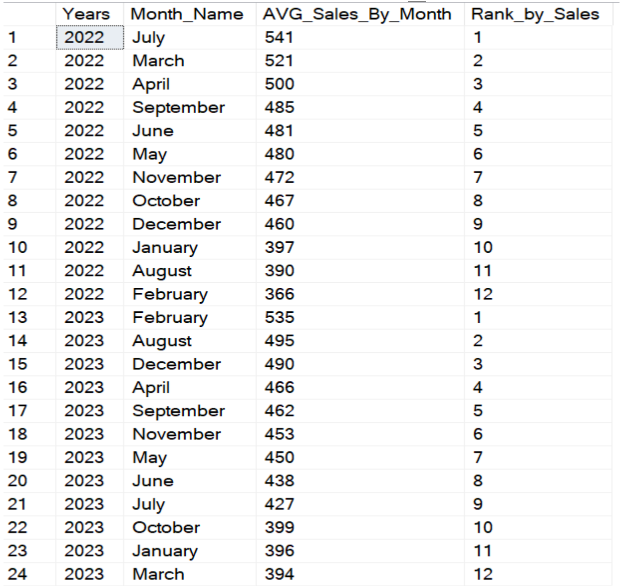
LAPTOP-MQQSV87C\SQLEXPRESS ... | LAPTOP-MQQSV87C\sahil ... | Retail_Sales | 00:00:00 | 306 rows

```sql
-- Q.6 Write a SQL query to find the total number of transactions
(transaction_id) made by each gender in each category.
SELECT
category,
gender as Gender,
COUNT(*) Total_Transaction
FROM Retail_Sales_Data
GROUP BY category, gender
ORDER BY 1
```



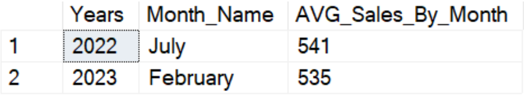
	category	Gender	Total_Transaction
1	Beauty	Female	330
2	Beauty	Male	281
3	Clothing	Female	347
4	Clothing	Male	351
5	Electronics	Female	335
6	Electronics	Male	343

```sql
-- Q.7 Write a SQL query to calculate the average sale for each month.
Find out best selling month in each year

SELECT
FORMAT(sale_date,'yyyy') as Years,
FORMAT(sale_date,'MMMM') Month_Name,
AVG(CAST(total_sale AS INT)) AS AVG_Sales_By_Month,
RANK() OVER(PARTITION BY FORMAT(sale_date,'yyyy') ORDER BY
AVG(CAST(total_sale AS INT)) DESC) AS Rank_by_Sales
FROM Retail_Sales_Data
GROUP BY FORMAT(sale_date,'yyyy'),
            FORMAT(sale_date,'MMMM')
```



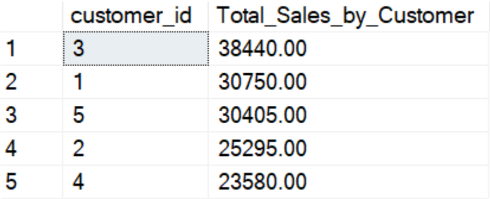
	Years	Month_Name	AVG_Sales_By_Month	Rank_by_Sales
1	2022	July	541	1
2	2022	March	521	2
3	2022	April	500	3
4	2022	September	485	4
5	2022	June	481	5
6	2022	May	480	6
7	2022	November	472	7
8	2022	October	467	8
9	2022	December	460	9
10	2022	January	397	10
11	2022	August	390	11
12	2022	February	366	12
13	2023	February	535	1
14	2023	August	495	2
15	2023	December	490	3
16	2023	April	466	4
17	2023	September	462	5
18	2023	November	453	6
19	2023	May	450	7
20	2023	June	438	8
21	2023	July	427	9
22	2023	October	399	10
23	2023	January	396	11
24	2023	March	394	12

```sql
-- Best Selling Month in Each Year
SELECT
Years,
Month_Name,
AVG_Sales_By_Month
FROM    (SELECT
           FORMAT(sale_date,'yyyy') as Years,
           FORMAT(sale_date,'MMMM') Month_Name,
           AVG(CAST(total_sale AS INT)) AS AVG_Sales_By_Month,
           RANK() OVER(PARTITION BY FORMAT(sale_date,'yyyy') ORDER BY
AVG(CAST(total_sale AS INT)) DESC) AS Rank_by_Avg_Sales
           FROM Retail_Sales_Data
           GROUP BY FORMAT(sale_date,'yyyy'),
                       FORMAT(sale_date,'MMMM') )t

WHERE Rank_by_Avg_Sales = 1
```



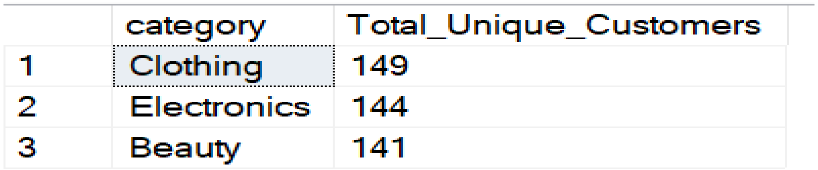
	Years	Month_Name	AVG_Sales_By_Month
1	2022	July	541
2	2023	February	535

```
-- Q.8 Write a SQL query to find the top 5 customers based on the
highest total sales

SELECT TOP 5 customer_id,
SUM(total_sale) as Total_Sales_by_Customer
FROM Retail_Sales_Data
GROUP BY customer_id
ORDER BY 2 DESC
```



	customer_id	Total_Sales_by_Customer
1	3	38440.00
2	1	30750.00
3	5	30405.00
4	2	25295.00
5	4	23580.00

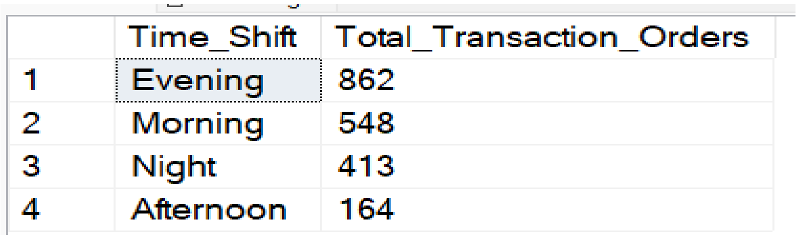
```
-- Q.9 Write a SQL query to find the number of unique customers who
purchased items from each category.

SELECT
category,
COUNT(DISTINCT customer_id) Total_Unique_Customers
FROM Retail_Sales_Data
GROUP BY category
ORDER BY 2 DESC
```



	category	Total_Unique_Customers
1	Clothing	149
2	Electronics	144
3	Beauty	141

```
-- Q.10 Write a SQL query to create each shift and number of orders
(Example Morning <=12, Afternoon Between 12 & 17, Evening >17 and rest
is of Night)

SELECT
    CASE
        WHEN DATEPART(HH,sale_time) < 12 THEN 'Morning'
        WHEN DATEPART(HH,sale_time) >= 12 AND DATEPART(HH,sale_time) <
17 THEN 'Afternoon'
        WHEN DATEPART(HH,sale_time) >= 17 AND DATEPART(HH,sale_time) <
21 THEN 'Evening'
        ELSE 'Night'
    END AS Time_Shift, COUNT(*) Total_Transaction_Orders
FROM Retail_Sales_Data
Group  by
```

```
      CASE
                 WHEN DATEPART(HH,sale_time) < 12 THEN 'Morning'
                 WHEN DATEPART(HH,sale_time) >= 12 AND
DATEPART(HH,sale_time) < 17 THEN 'Afternoon'
                 WHEN DATEPART(HH,sale_time) >= 17 AND
DATEPART(HH,sale_time) < 21 THEN 'Evening'
                 ELSE 'Night'
            END
ORDER BY 2 DESC
```



	Time_Shift	Total_Transaction_Orders
1	Evening	862
2	Morning	548
3	Night	413
4	Afternoon	164