

# SQL Query Project Documentation

**Prepared by:** Aparna Sharmaa

**Role:** Data Scientist/Data Analyst (Fresher)

**Date:** January 2025

## Project Overview:

This project focuses on analyzing retail sales data using SQL to extract business insights. The dataset includes details such as Invoice Number, Product Details, Quantity, Price, Customer Information, and Country.

### Key skills demonstrated in this project:

- ☐ Data cleaning and preprocessing
- ☐ Writing SQL queries for descriptive analytics
- ☐ Using aggregation, window functions, and conditional logic
- ☐ Preparing data insights for visualization tools like Power BI

## Dataset Summary

Column Name	Description
InvoiceNo	Unique transaction ID
StockCode	Product code
Description	Product name
Quantity	Number of units purchased
InvoiceDate	Date and time of transaction
UnitPrice	Price per product unit
CustomerID	Unique customer ID
Country	Customer's country

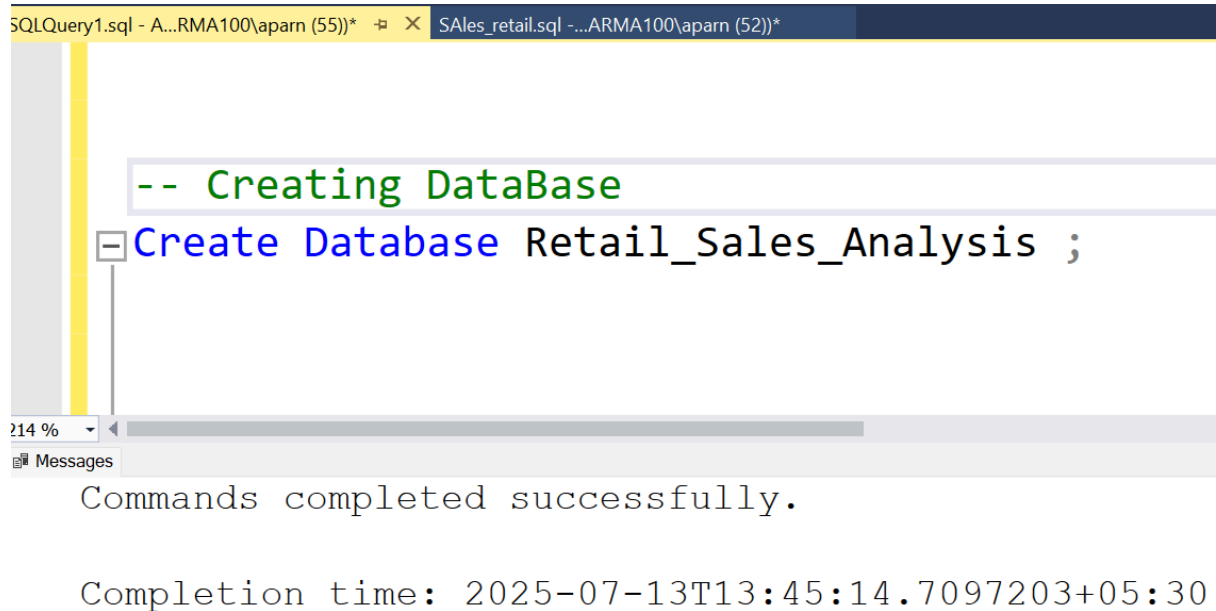
## Tools Used

**Microsoft SQL Server**

**Power BI**

## Create Database:

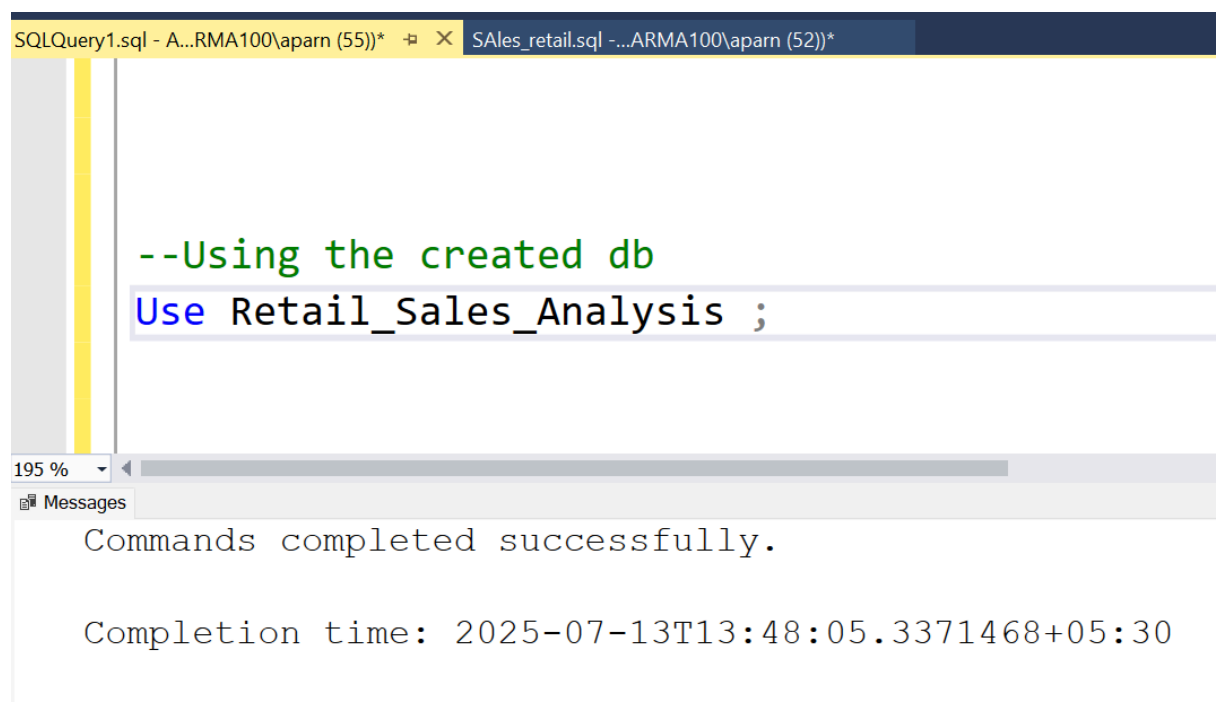
Create Database Retail\_Sales\_Analysis ;



The screenshot shows a SQL Server Enterprise Manager window with two tabs: 'SQLQuery1.sql - A...RMA100\aparn (55))\*' and 'Sales\_retail.sql - ...ARMA100\aparn (52))\*'. The 'Sales\_retail.sql' tab is active, displaying the following SQL command:   
`-- Creating DataBase`  
`Create Database Retail_Sales_Analysis ;`  
Below the command editor, the 'Messages' pane shows the output:   
Commands completed successfully.  
Completion time: 2025-07-13T13:45:14.7097203+05:30

## To use the database:

Use Retail\_Sales\_Analysis ;



The screenshot shows the same SQL Server Enterprise Manager window. The 'Sales\_retail.sql' tab is active, displaying the following SQL command:   
`--Using the created db`  
`Use Retail_Sales_Analysis ;`  
Below the command editor, the 'Messages' pane shows the output:   
Commands completed successfully.  
Completion time: 2025-07-13T13:48:05.3371468+05:30

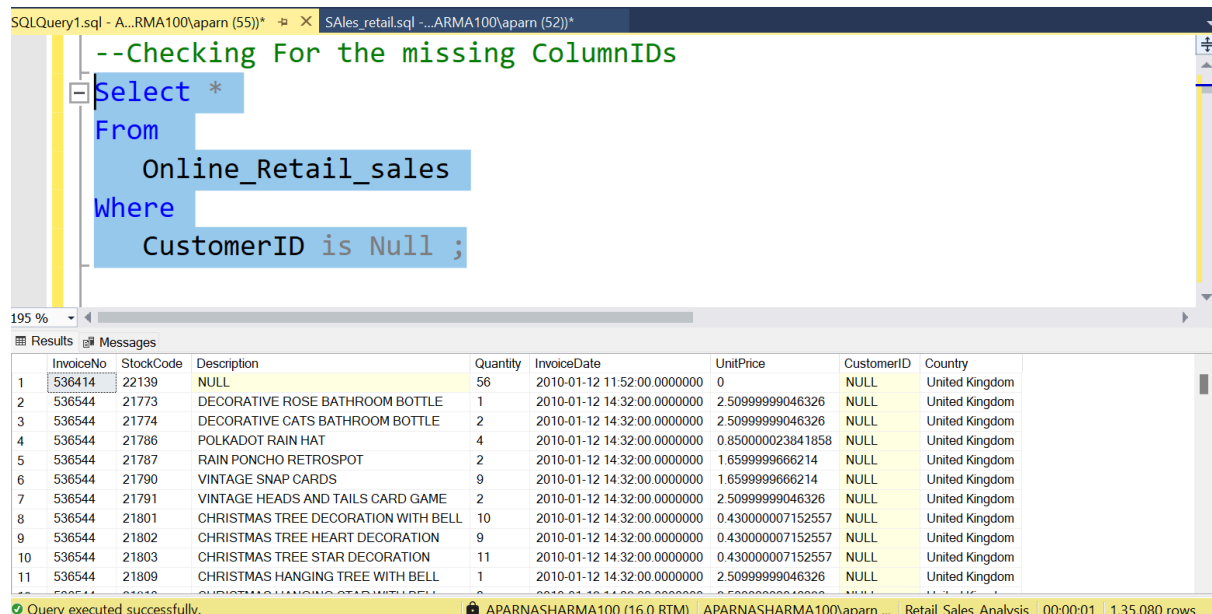
## Data Cleaning Steps

Before performing sales analysis, the following SQL data cleaning steps were applied to ensure accuracy and reliability:

### 1. Remove Rows with Missing Customer IDs

Firstly I checked for the rows with missing customerIds using :

```
Select *  
From  
    Online_Retail_sales  
Where  
    CustomerID is Null ;
```



The screenshot shows a SQL query window with the following text:

```
--Checking For the missing ColumnIDs  
Select *  
From  
    Online_Retail_sales  
Where  
    CustomerID is Null ;
```

Below the query window, the 'Results' pane displays a table with the following data:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536414	22139	NULL	56	2010-01-12 11:52:00.0000000	0	NULL	United Kingdom
2	536544	21773	DECORATIVE ROSE BATHROOM BOTTLE	1	2010-01-12 14:32:00.0000000	2.50999999046326	NULL	United Kingdom
3	536544	21774	DECORATIVE CATS BATHROOM BOTTLE	2	2010-01-12 14:32:00.0000000	2.50999999046326	NULL	United Kingdom
4	536544	21786	POLKADOT RAIN HAT	4	2010-01-12 14:32:00.0000000	0.850000023841858	NULL	United Kingdom
5	536544	21787	RAIN PONCHO RETROSPOT	2	2010-01-12 14:32:00.0000000	1.6599999666214	NULL	United Kingdom
6	536544	21790	VINTAGE SNAP CARDS	9	2010-01-12 14:32:00.0000000	1.6599999666214	NULL	United Kingdom
7	536544	21791	VINTAGE HEADS AND TAILS CARD GAME	2	2010-01-12 14:32:00.0000000	2.50999999046326	NULL	United Kingdom
8	536544	21801	CHRISTMAS TREE DECORATION WITH BELL	10	2010-01-12 14:32:00.0000000	0.430000007152557	NULL	United Kingdom
9	536544	21802	CHRISTMAS TREE HEART DECORATION	9	2010-01-12 14:32:00.0000000	0.430000007152557	NULL	United Kingdom
10	536544	21803	CHRISTMAS TREE STAR DECORATION	11	2010-01-12 14:32:00.0000000	0.430000007152557	NULL	United Kingdom
11	536544	21809	CHRISTMAS HANGING TREE WITH BELL	1	2010-01-12 14:32:00.0000000	2.50999999046326	NULL	United Kingdom

At the bottom of the screenshot, a status bar indicates: 'Query executed successfully. APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail Sales Analysis 00:00:01 1,35,080 rows'.

Rows without customer IDs were considered incomplete and removed.

```
Delete From  
    Online_Retail_sales  
Where  
    CustomerID is Null ;
```

```

SQLQuery1.sql - A...RMA100\aparn (55))* X Sales_retail.sql -...ARMA100\aparn (52))*
-- Deleting these null Columnids
Delete From
    Online_Retail_sales
Where
    CustomerID is Null ;

(135080 rows affected)

Completion time: 2025-07-13T13:50:18.0564372+05:30

Query executed successfully.
APARNASHARMA100 (16.0 RTM) | APARNASHARMA100\aparn ... | Retail_Sales_Analysis | 00:00:00 | 0 rows

```

## 2. Handle Negative Quantity Values

Rows with negative quantity and unit price were identified .

```

Select *
From
    Online_Retail_sales
Where
    Quantity < 0 ;

```

--For Unit Price:

```

Select *
From
    Online_Retail_sales
Where
    UnitPrice < 0;

```

```

SQLQuery1.sql - A...RMA100\aparn (55))* X Sales_retail.sql -...ARMA100\aparn (52))*
--Checking for the Negative Quantity values And Unit Price:
Select
    *
From
    Online_Retail_sales
Where
    Quantity < 0 ;
--For Unit Price:
Select
    *
From
    Online_Retail_sales
Where
    UnitPrice < 0;

Results
InvoiceNo  StockCode  Description  Quantity  InvoiceDate  UnitPrice  CustomerID  Country
1  C536642  21463  MIRRORRED DISCO BALL  -1  2010-02-12 11:56:00.0000000  5.94999980926514  14390  United Kingdom
2  C536734  22780  LIGHT GARLAND BUTTERFILES PINK  -4  2010-02-12 12:50:00.0000000  4.25  16042  United Kingdom
3  C536734  85048  15CM CHRISTMAS GLASS BALL 20 LIGHTS  -1  2010-02-12 12:50:00.0000000  7.94999980926514  16042  United Kingdom
4  C536737  22798  ANTIQUE GLASS DRESSING TABLE POT  -2  2010-02-12 13:05:00.0000000  2.95000004768372  15922  United Kingdom
5  C536763  94547  DOTA TMS GII VED AMSEL S T LIGHT UP ND  -0.260  2010-03-15 14:23:00.0000000  0.000000000000000  16042  United Kingdom

Query executed successfully.
APARNASHARMA100 (16.0 RTM) | APARNASHARMA100\aparn ... | Retail_Sales_Analysis | 00:00:00 | 8,905 rows

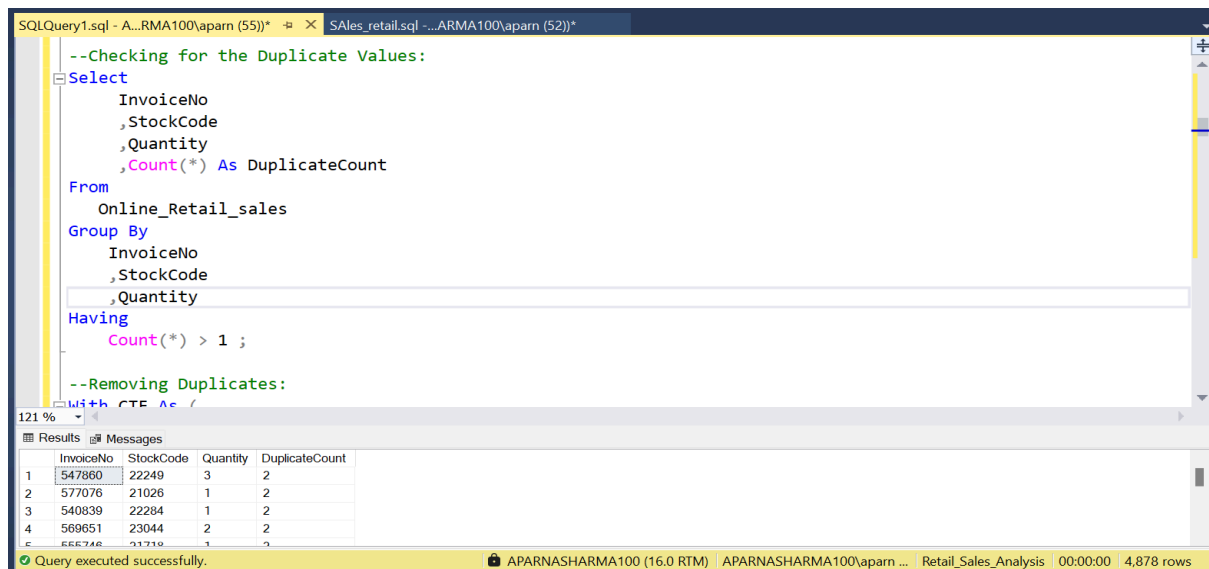
```

### 3.Remove Duplicate Records

Duplicate records were identified based on InvoiceNo, StockCode, and Quantity.  
Duplicates were removed using SQL's window function ROW\_NUMBER():

#### --Checking for the Duplicate Values:

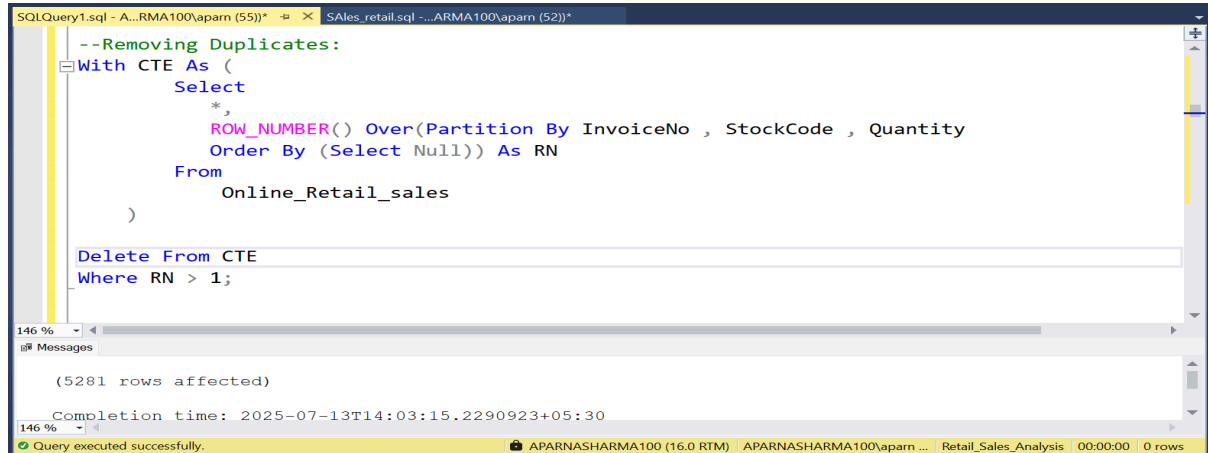
```
Select
    InvoiceNo
    ,StockCode
    ,Quantity
    ,Count(*) As DuplicateCount
From
    Online_Retail_sales
Group By
    InvoiceNo
    ,StockCode
    ,Quantity
Having
    Count(*) > 1 ;
```



#### --Removing Duplicates:

```
With CTE As (
    Select
        *
        ,ROW_NUMBER() Over(Partition By InvoiceNo , StockCode , Quantity
        Order By (Select Null)) As RN
    From
        Online_Retail_sales
)
-- Removing Duplicates:
Delete From CTE
Where RN > 1
```

Delete From CTE  
Where RN > 1;



```
--Removing Duplicates:
With CTE As (
    Select
        *,
        ROW_NUMBER() Over(Partition By InvoiceNo , StockCode , Quantity
        Order By (Select Null)) As RN
    From
        Online_Retail_sales
)

Delete From CTE
Where RN > 1;
```

(5281 rows affected)

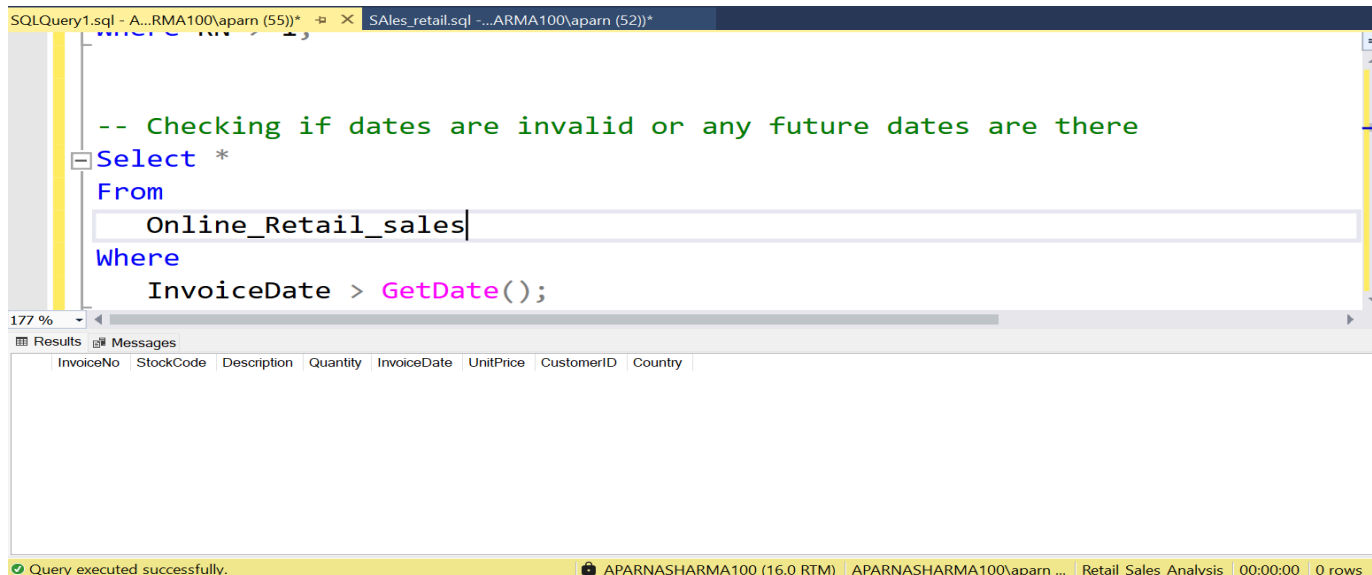
Completion time: 2025-07-13T14:03:15.2290923+05:30

Query executed successfully.

## 4. Validate Invoice Date

Dates were checked to ensure there were no future-dated transactions:

```
SELECT *
FROM
    Online_Retail_sales
WHERE
    InvoiceDate > GETDATE();
```



```
-- Checking if dates are invalid or any future dates are there
Select *
From
    Online_Retail_sales
Where
    InvoiceDate > GetDate();
```

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
-----------	-----------	-------------	----------	-------------	-----------	------------	---------

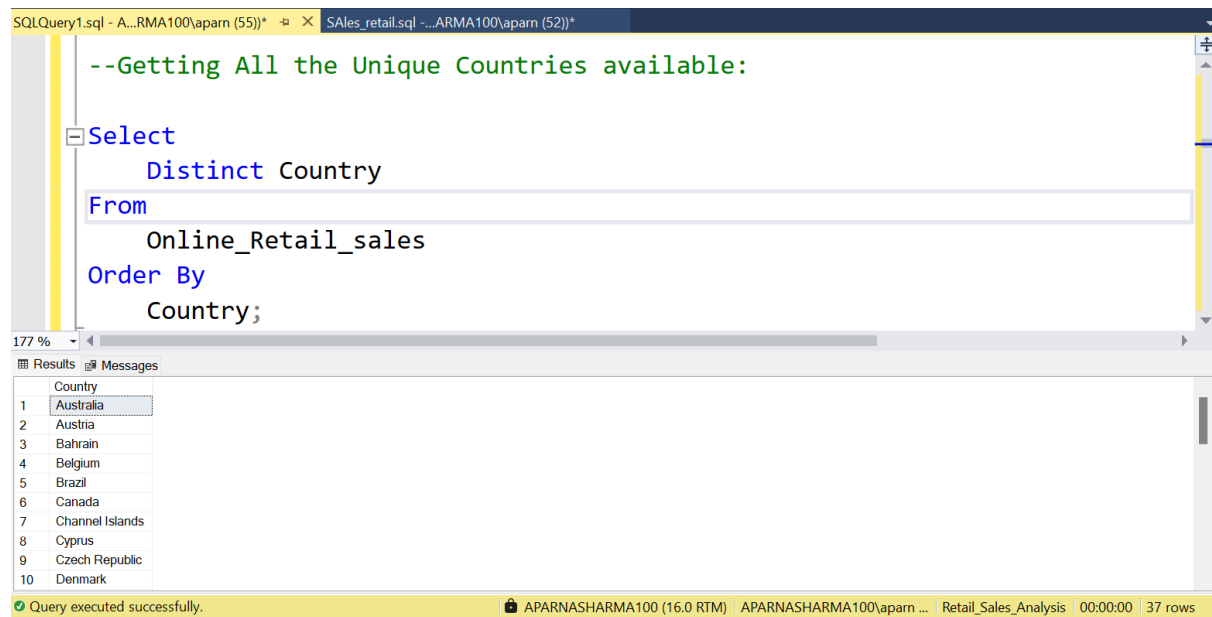
Query executed successfully.

## 6. Standardize Country Names

All unique country values were reviewed for consistency:

```
SELECT
    DISTINCT Country
FROM
```

Online\_Retail\_sales  
ORDER BY  
Country;



### Note:

These cleaning steps helped in preparing clean and reliable sales data for further analysis using SQL queries, Power BI dashboards.

## 3 SQL Business Analysis Queries

The following queries were performed on the cleaned dataset to extract key business insights.

### 1. Total Sales Revenue by Country:

```
Select  
    Country  
    ,Round(Sum(UnitPrice * Quantity),2)  
    As TotalSalesRevenue  
From  
    Online_Retail_sales  
Group By  
    Country  
Order By  
    TotalSalesRevenue Desc;
```

SQLQuery1.sql - A...RMA100\aparn (55))\* X Sales\_retail.sql -...ARMA100\aparn (52))\*

```
-- What is total sales revenue by country?
Select
Country
, Round(Sum(UnitPrice * Quantity), 2)
As TotalSalesRevenue
From
Online_Retail_sales
Group By
Country
Order By
TotalSalesRevenue Desc;
```

133 %

Results Messages

	Country	TotalSalesRevenue
1	United Kingdom	6748546.26
2	Netherlands	284661.54
3	EIRE	250001.78
4	Germany	220960.13
5	France	196626.05
6	Australia	137009.77
7	Switzerland	55739.4
8	Spain	54756.03
9	Belgium	40910.96
10	Sweden	36585.41

Query executed successfully. APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail\_Sales\_Analysis 00:00:00 37 rows

### Insight:

Identifies which countries contribute the most to the business's revenue.

**Top 5 are :** UK , Netherlands , EIRE , Germany and France.

## 2. Total Number of Unique Customers:

Select

Count(Distinct CustomerID)  
As Total\_Unique\_Customers

From

Online\_Retail\_sales;

SQLQuery1.sql - A...RMA100\aparn (55))\* X Sales\_retail.sql -...ARMA100\aparn (52))\*

```
-- Find the Total number of Unique Customer:
Select
Count(Distinct CustomerID)
As Total_Unique_Customers
From
Online_Retail_sales
```

177 %

Results Messages

	Total_Unique_Customers
1	4372

Top 10 Countries by Sales:

### Insight:

Helps measure the active customer base.

## 3. Top 10 Countries by Sales :

Select

Top 10 Country  
, Sum(Quantity\*UnitPrice) As



```

Revenue
From
Online_Retail_sales
Group By
Country
Order By
Revenue Desc;

```

The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor contains the following SQL code:

```

-- Top 10 Countries by Sales:
Select
    Top 10 Country
    ,Sum(Quantity*UnitPrice) As
    Revenue
From
    Online_Retail_sales
Group By
    Country
Order By
    Revenue Desc;

```

The results pane displays the following data:

	Country	Revenue
1	United Kingdom	6748546.26286513
2	Netherlands	284661.539452553
3	EIRE	250001.778775401
4	Germany	220960.128893398
5	France	196626.048984453

The status bar at the bottom indicates: Query executed successfully. APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail\_Sales\_Analysis 00:00:00 10 rows

**Insight:**  
Useful for targeting marketing or business expansion.

#### 4. Top 10 Products Sold :

```

Select
    Top 10 Description
    ,Sum(Quantity) As
    Total_Quantity_Sold
From
    Online_Retail_sales
Group By
    Description
    ,Quantity
Order By
    Quantity Desc;

```

SQLQuery1.sql - A...RMA100\aparn (55))\* X Sales\_retail.sql -...ARMA100\aparn (52))\*

```
-- What are the top 10 products sold?
Select
    Top 10 Description
    ,Sum(Quantity) As
    Total_Quantity_Sold
From
    Online_Retail_sales
Group By
    Description
Order By
    Quantity Desc;
```

146 %

	Description	Total_Quantity_Sold
1	PAPER CRAFT, LITTLE BIRDIE	80995
2	MEDIUM CERAMIC TOP STORAGE JAR	74215
3	ASSTD DESIGN 3D PAPER STICKERS	12540
4	WORLD WAR 2 GLIDERS ASSTD DESIGNS	4800
5	SMALL POPCORN HOLDER	4300

Query executed successfully. APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail\_Sales\_Analysis 00:00:00 10 rows

### Insight:

Shows best-selling products.

## 5. Customer Segmentation: Most Frequent Buyers

```
SELECT
    TOP 5
    CustomerID,
    COUNT(*) AS PurchaseCount
FROM
    Online_Retail_sales
WHERE
    CustomerID IS NOT NULL
GROUP BY
    CustomerID
ORDER BY
    PurchaseCount DESC;
```

SQLQuery1.sql - A...RMA100\aparn (55))\* X Sales\_retail.sql -...ARMA100\aparn (52))\*

```
--Customer Segmentation: (most frequent buyers):
SELECT
    TOP 5
    CustomerID,
    COUNT(*) AS PurchaseCount
FROM
    Online_Retail_sales
WHERE
    CustomerID IS NOT NULL
GROUP BY
    CustomerID
ORDER BY
    PurchaseCount DESC;
```

146 %

	CustomerID	PurchaseCount
1	17841	7811
2	14911	5898
3	14096	5126
4	12748	4457
5	14606	2759

Query executed successfully. APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail\_Sales\_Analysis 00:00:00 5 rows

### Insight:

Helps identify loyal or high-frequency customers.

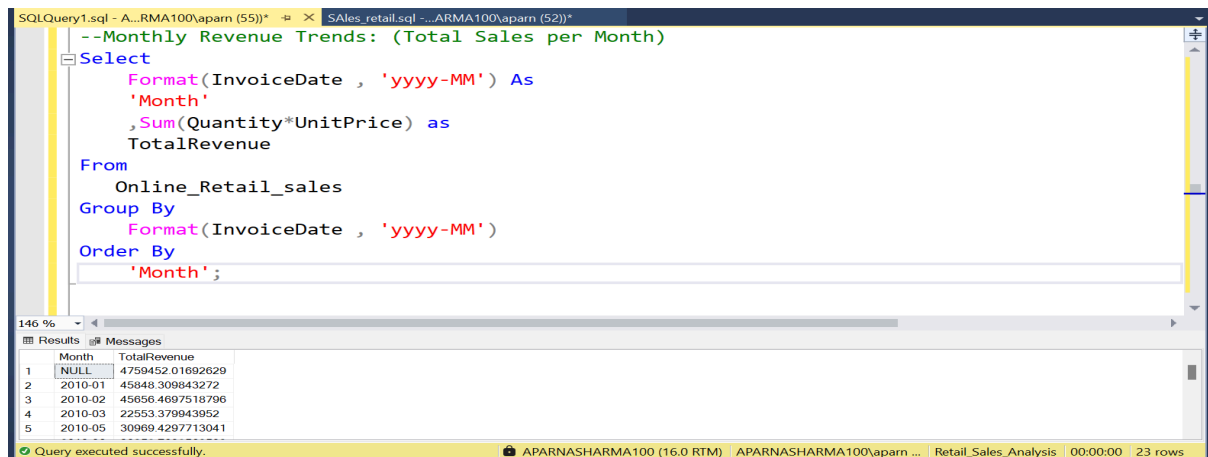
## 6. Monthly Revenue Trends

```
Select
    Format(InvoiceDate , 'yyyy-MM') As
```

```

        'Month'
        ,Sum(Quantity*UnitPrice) as
        TotalRevenue
From
    Online_Retail_sales
Group By
    Format(InvoiceDate , 'yyyy-MM')
Order By
    'Month';

```



### Insight:

Helps spot seasonal sales patterns.

## 7. Average Basket Value Per Customer

```

Select
    CustomerID
    ,Round(Avg(Quantity*UnitPrice),2) As
    AverageBasketVaue
From
    Online_Retail_sales
Where
    CustomerID is not Null
Group By
    CustomerID
Order By
    AverageBasketVaue Desc;

```

SQLQuery1.sql - A...RMA100\aparn (55)) \* Sales\_retail.sql - ...ARMA100\aparn (52)) \*

```
-- Find the average basket value per customer:
Select
    CustomerID
    ,Round(Avg(Quantity*UnitPrice),2) As
    AverageBasketVaue
From
    Online_Retail_sales
Where
    CustomerID is not Null
Group By
    CustomerID
Order By
    AverageBasketVaue Desc;
```

146 %

Results Messages

	CustomerID	AverageBasketVaue
1	15195	3861
2	13135	3096
3	17846	2033.1
4	16532	1687.2
5	15749	1435.73

Query executed successfully. APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail\_Sales\_Analysis 00:00:00 4,372 rows

### Insight:

Useful for understanding customer spending habits.

## 8. Products Sold in More Than 500 Invoices

Select

Description

,Count(Distinct InvoiceNo)

As InvoiceCount

From

Online\_Retail\_sales

Group By

Description

Having

Count(Distinct InvoiceNo) > 500

Order By

InvoiceCount Desc;

SQLQuery1.sql - A...RMA100\aparn (55)) \* Sales\_retail.sql - ...ARMA100\aparn (52)) \*

```
--Products Sold in more than 500 invoices:
Select
    Description
    ,Count(Distinct InvoiceNo)
    As InvoiceCount
From
    Online_Retail_sales
Group By
    Description
Having
    Count(Distinct InvoiceNo) > 500
Order By
    InvoiceCount Desc;
```

146 %

Results Messages

	Description	InvoiceCount
1	WHITE HANGING HEART T-LIGHT HOLDER	2013
2	REGENCY CAKESTAND 3 TIER	1884
3	JUMBO BAG RED RETROSPOT	1643
4	PARTY BUNTING	1399
5	ASSORTED COLOUR BIRD ORNAMENT	1385

Query executed successfully. APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail\_Sales\_Analysis 00:00:08 123 rows

### Insight:

Identifies consistently popular products.

## 9. Top 3 Products by Revenue Per Country

With ProductRevenue As (

Select

Country

,Description

,Sum(Quantity \* UnitPrice)

As Revenue

,ROW\_NUMBER() Over(Partition By Country

Order By Sum(Quantity \* UnitPrice)Desc)

As Ranked

From

Online\_Retail\_sales

Group By

Country

,Description

)

Select

Country

,Description

,Revenue

From

ProductRevenue

Where

Ranked <= 3

Order By

Country

,Revenue Desc;

The screenshot shows a SQL query window with the following text:

```
--What are Top-3 Products on the basis of revenue per country ?
With ProductRevenue As (
    Select
        Country
        ,Description
        ,Sum(Quantity * UnitPrice)
        As Revenue
        ,ROW_NUMBER() Over(Partition By Country
        Order By Sum(Quantity * UnitPrice)Desc)
        As Ranked
    From
        Online_Retail_sales
    Group By
        Country
        ,Description
)

Select
    Country
    ,Description
    ,Revenue
From
    ProductRevenue
Where
    Ranked <= 3
Order By
    Country
    ,Revenue Desc;
```

Below the query window, the 'Results' tab is active, displaying a table with 5 rows and 3 columns: Description, InvoiceCount, and Revenue. The data is as follows:

	Description	InvoiceCount	Revenue
1	WHITE HANGING HEART T-LIGHT HOLDER	2013	1884
2	REGENCY CAKESTAND 3 TIER	1884	1643
3	JUMBO BAG RED RETROSPOT	1643	1399
4	PARTY BUNTING	1399	1385
5	ASSORTED COLOUR BIRD ORNAMENT	1385	

The status bar at the bottom indicates: Query executed successfully. APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail\_Sales\_Analysis | 00:00:08 | 123 rows

**Insight:**

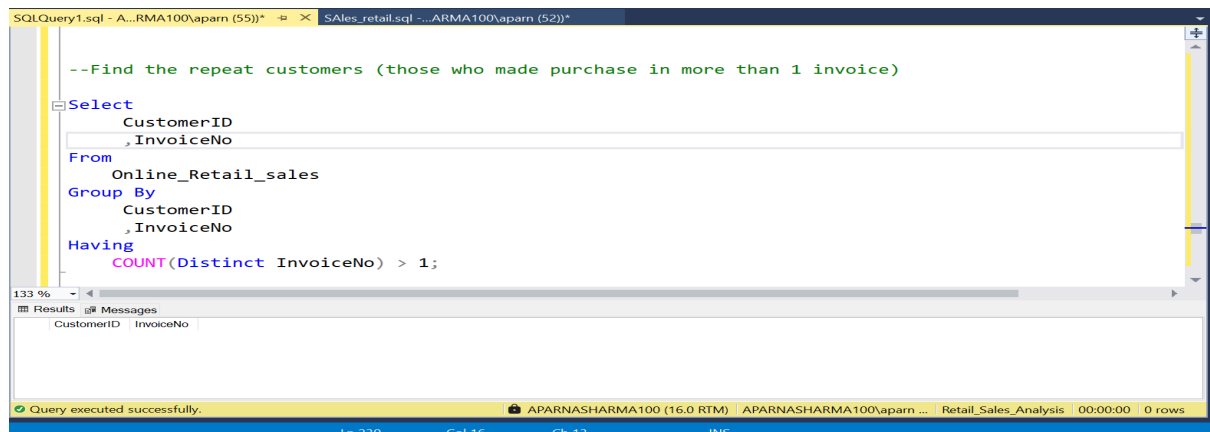
Helps prioritize product distribution and inventory by region.

## 10. Repeat Customers

```

Select
    CustomerID
    ,InvoiceNo
From
    Online_Retail_sales
Group By
    CustomerID
    ,InvoiceNo
Having
    COUNT(Distinct InvoiceNo) > 1;

```



### Insight:

Helps measure customer retention rate.

## 11. Flag Transactions with Negative Quantity (Returns or Refunds)

```

Select
    *,
    Case
        When Quantity < 0 Then 'Return / Refund'
        Else 'Normal Sales'
    End as
        'Transaction_Type'
From
    Online_Retail_sales;

```

SQLQuery1.sql - A...RMA100\aparn (55)\*    Sales-retail.sql -...ARMA100\aparn (52)\*

```
--Flag all the transactions where the quantity is negative (possible returns or refunds):
Select
    *,
    Case
        When Quantity < 0 Then 'Return / Refund'
        Else 'Normal Sales'
    End as
        'Transaction_Type'
From
    Online_Retail_sales;
```

146 %    Results    Messages

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country	Transaction_Type
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-01-12 08:26:00.0000000	2.54999995231628	17850	United Kingdom	Normal Sales
2	536365	71053	WHITE METAL LANTERN	6	2010-01-12 08:26:00.0000000	3.39000010490417	17850	United Kingdom	Normal Sales
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-01-12 08:26:00.0000000	2.75	17850	United Kingdom	Normal Sales
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-01-12 08:26:00.0000000	3.39000010490417	17850	United Kingdom	Normal Sales
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-01-12 08:26:00.0000000	3.39000010490417	17850	United Kingdom	Normal Sales

Query executed successfully.    APARNASHARMA100 (16.0 RTM)    APARNASHARMA100\aparn ...    Retail\_Sales\_Analysis    00:00:03    4,01,548 rows

## 12. Product Ranking by Revenue Within Each Country

With ProductRevenue As (

```
    Select
        Description
        ,Country
        ,Sum(Quantity * UnitPrice)
        As TotalRevenue
    From
        Online_Retail_sales
    Group By
        Country
        ,Description
    )
```

Select

```
    Description
    ,Country
    ,TotalRevenue
    ,Rank() Over(
        Partition By Country
        Order By TotalRevenue Desc)
    As Ranked
```

From

ProductRevenue;

SQLQuery1.sql - A...RMA100\aparn (55)\* < > Sales\_retail.sql - ...ARMA100\aparn (52)\*

```
--For each product , rank it within its country based on total revenue using window functions:
With ProductRevenue As (
    Select
        Description
        ,Country
        ,Sum(Quantity * UnitPrice)
        As TotalRevenue
    From
        Online_Retail_sales
    Group By
        Country
        ,Description
)
Select
    Description
    ,Country
    ,TotalRevenue
    ,Rank() Over(
        Partition By Country
        Order By TotalRevenue Desc)
    As Ranked
From
    ProductRevenue;
```

83 %

Results Messages

	Description	Country	TotalRevenue	Ranked
1	RED VINTAGE SPOT BEAKER	Austria	6.80000019073486	292
2	BLUE VINTAGE SPOT BEAKER	Austria	6.80000019073486	292
3	PINK VINTAGE SPOT BEAKER	Austria	6.80000019073486	292
4	RED RETROSPOT CUP	Austria	6.80000019073486	292
5	PINK POLKADOT CUP	Austria	6.80000019073486	292

Query executed successfully.

APARNASHARMA100 (16.0 RTM) APARNASHARMA100\aparn ... Retail\_Sales\_Analysis 00:00:00 19,351 rows

### Insight:

Provides product popularity ranking for regional business strategy.

## Conclusion:

### Project Summary and Key Learnings

This project involved cleaning and analyzing retail sales data using Microsoft SQL Server.

#### Key insights derived:

- ❖ Revenue distribution by country and product
- ❖ Customer segmentation: identifying top buyers and repeat customers
- ❖ Seasonal sales trends
- ❖ Return/refund detection using SQL case logic
- ❖ Product ranking with window functions

#### Skills demonstrated:

- SQL aggregation and grouping
- Window functions: ROW\_NUMBER, RANK
- Data cleaning using SQL (removing duplicates, handling missing values)
- Business insights preparation for dashboards (Power BI)

### References :

Dataset: Online Retail Dataset (Publicly available version from Kaggle/UCI Repository)



SQL Server Documentation

Power BI Official Guide.