1. BASIC ANALYSIS

1. Find the total revenue (sum of Quantity * UnitPrice) generated from all invoices.

```
-- 1 Find the total revenue (sum of Quantity * UnitPrice) generated from all invoices.

SELECT SUM(Quantity * UnitPrice) AS TotalRevenue
FROM retail;
```

Results | Chart

	TOTALREVENUE
1	9747747.93

2. . Count the number of unique products (StockCode) sold

```
-- 2. Count the number of unique products (StockCode) sold.

SELECT COUNT(DISTINCT StockCode) AS UniqueProducts
FROM retail;
```

Results | Chart

	UNIQUEPRODUCTS
1	4070

3. . Identify the total number of invoices in the dataset.

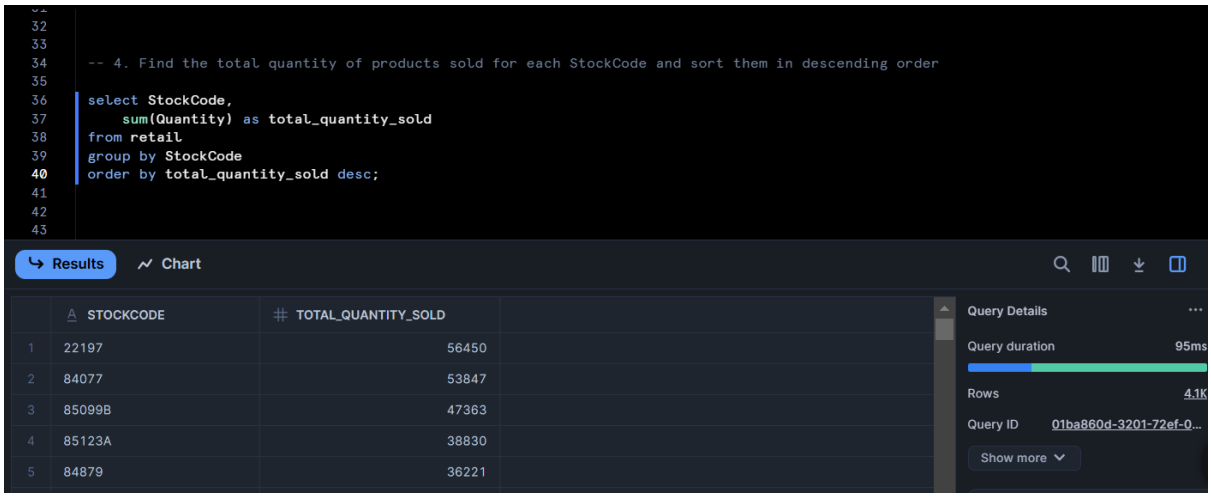
```
-- 3. Identify the total number of invoices in the dataset.

SELECT COUNT(DISTINCT InvoiceNo) AS TotalInvoices
FROM retail;
```

Results | Chart

	TOTALINVOICES
1	25900

4. Find the total quantity of products sold for each StockCode and sort them in descending order.



5. 5. Count the number of transactions (distinct InvoiceNo) per customer (CustomerID).

2. Customer Analysis

1. Identify the top 5 customers who have generated the highest revenue.

```
55
56
57  -- 1. Identify the top 5 customers who have generated the highest revenue.
58
59  Select
60      customerid,
61      sum (quantity * unitprice) as totalrevenue
62      from retail
63      group by customerid
64      order by totalrevenue desc
65      limit 5;
66
```

Results | Chart

	CUSTOMERID	TOTALREVENUE
1	14646	282038.74
2	18102	272390.60
3	17450	187523.77
4	14911	147375.15
5	14156	126339.52

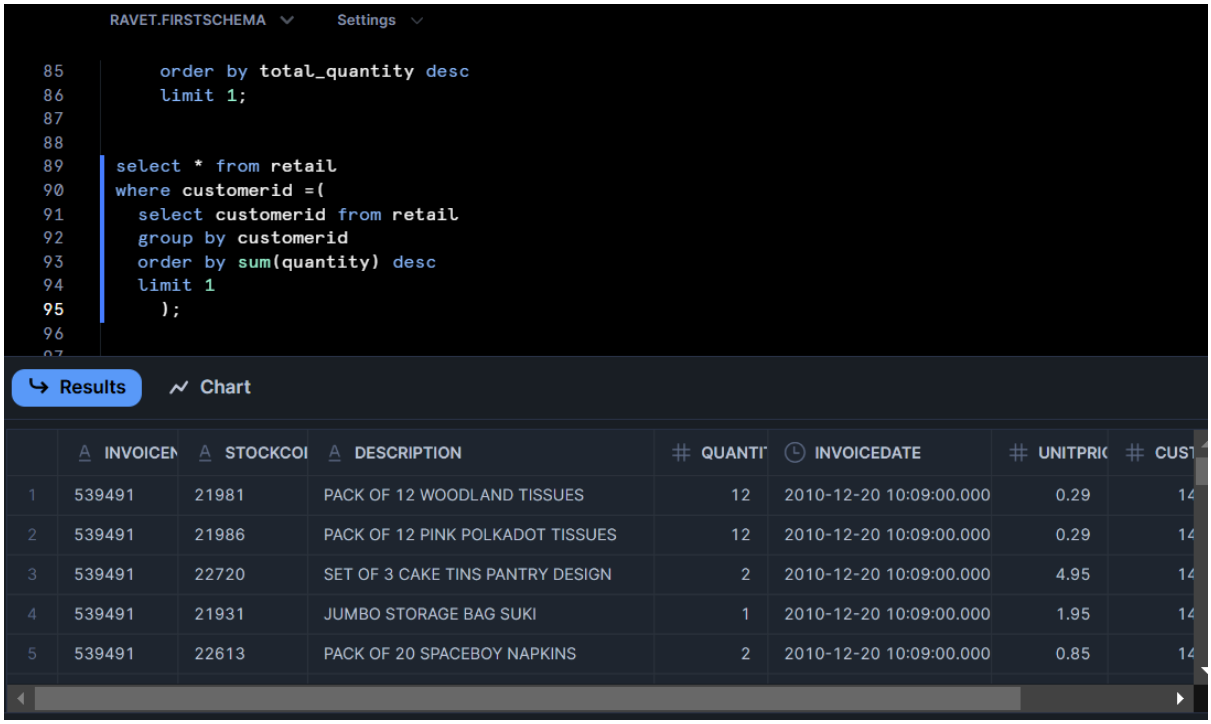
2. Find the average number of products purchased per customer.

```
66
67
68  -- 2 Find the average number of products purchased per customer.
69
70  SELECT
71      customerid,
72      AVG(Quantity) AS Average_qty
73  from retail
74  group by customerid
75  order by average_qty desc;
76
```

Results | Chart

	CUSTOMERID	AVERAGE_QTY
1	13135	4300.000000
2	16754	2140.000000
3	16308	2000.000000
4	15195	1404.000000
5	13256	1141.000000
6	14609	838.400000

3. Retrieve all transactions made by the customer who has purchased the most products in total.



RAVET.FIRSTSCHEMA Settings

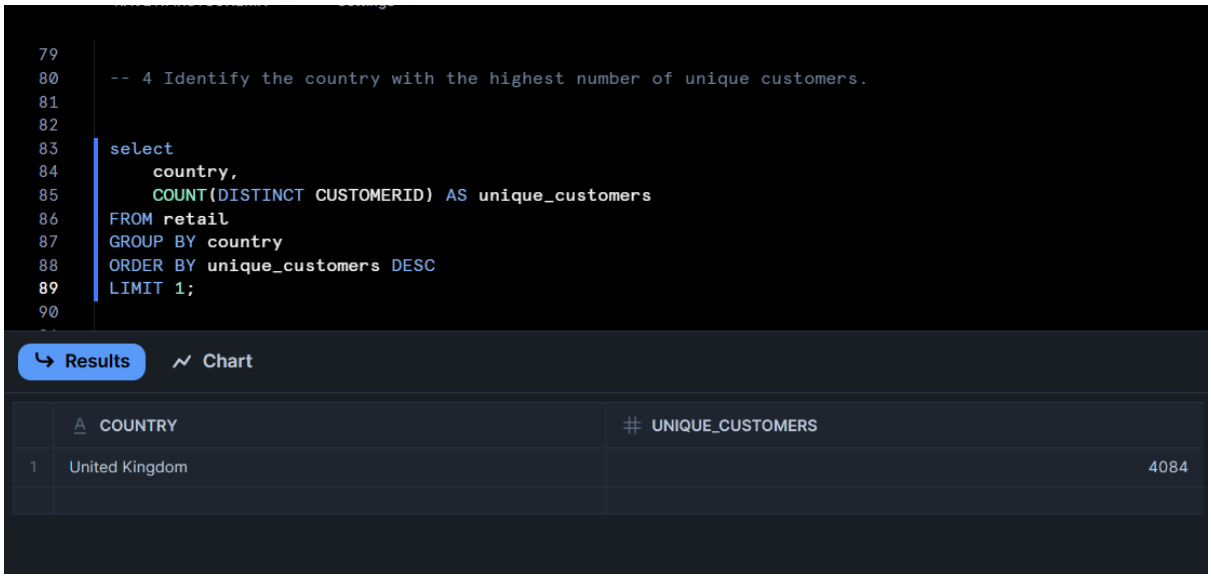
```
    order by total_quantity desc
    limit 1;


select * from retail
where customerid =(
  select customerid from retail
  group by customerid
  order by sum(quantity) desc
  limit 1
    );
```

Results Chart

	INVOICEN	STOCKCOI	DESCRIPTION	QUANTI	INVOICEDATE	UNITPRI(	CUST
1	539491	21981	PACK OF 12 WOODLAND TISSUES	12	2010-12-20 10:09:00.000	0.29	14
2	539491	21986	PACK OF 12 PINK POLKADOT TISSUES	12	2010-12-20 10:09:00.000	0.29	14
3	539491	22720	SET OF 3 CAKE TINS PANTRY DESIGN	2	2010-12-20 10:09:00.000	4.95	14
4	539491	21931	JUMBO STORAGE BAG SUKI	1	2010-12-20 10:09:00.000	1.95	14
5	539491	22613	PACK OF 20 SPACEBOY NAPKINS	2	2010-12-20 10:09:00.000	0.85	14

4. Identify the country with the highest number of unique customers.



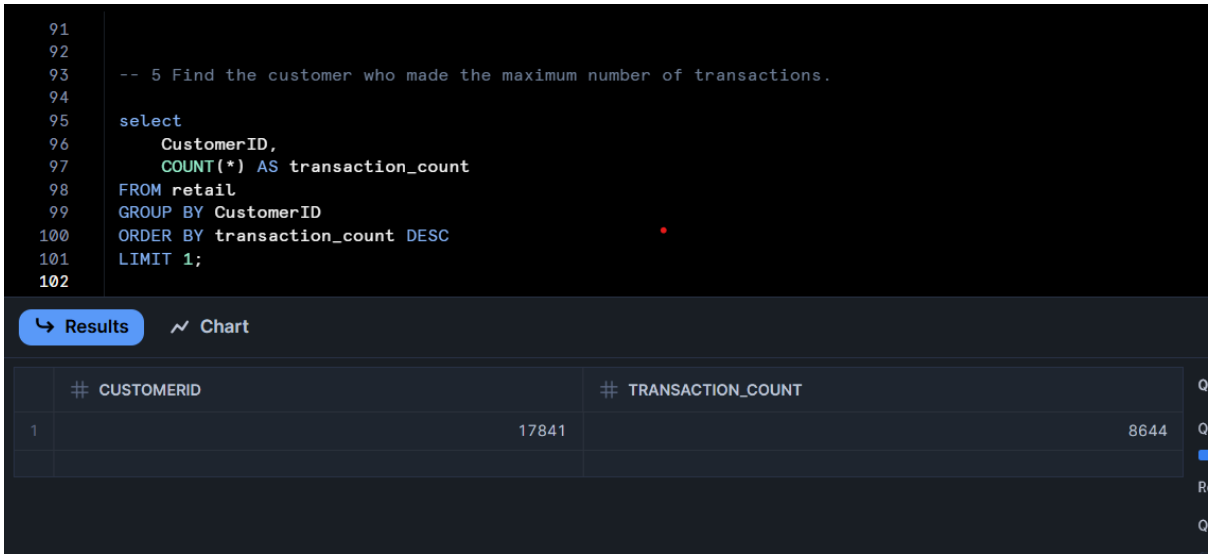
```
-- 4 Identify the country with the highest number of unique customers.


select
    country,
    COUNT(DISTINCT CUSTOMERID) AS unique_customers
FROM retail
GROUP BY country
ORDER BY unique_customers DESC
LIMIT 1;
```

Results Chart

	COUNTRY	UNIQUE_CUSTOMERS
1	United Kingdom	4084

5. Find the customer who made the maximum number of transactions



```
91
92
93  -- 5 Find the customer who made the maximum number of transactions.
94
95  select
96      CustomerID,
97      COUNT(*) AS transaction_count
98  FROM retail
99  GROUP BY CustomerID
100 ORDER BY transaction_count DESC
101 LIMIT 1;
102
```

Results | Chart

	CUSTOMERID	TRANSACTION_COUNT
1	17841	8644

3. Product-Based Analysis

1. List the top 5 most frequently purchased products (based on total quantity sold).

```sql
-- 1 List the top 5 most frequently purchased products (based on total quantity sold).

select
    stockcode,
    sum(quantity) as total_quantity
    from retail
    group by stockcode
    order by total_quantity desc
    limit 5;
```

Results | Chart

	STOCKCODE	TOTAL_QUANTITY
1	22197	56450
2	84077	53847
3	85099B	47363
4	85123A	38830
5	84879	36221

2. Find the product that generated the highest revenue.

```sql
-- 2 Find the product that generated the highest revenue.
select
    stockcode,
    sum(quantity * unitprice) as revenue
    from retail
    group by stockcode
    order by revenue desc
    limit  1;
```

Results | Chart

	STOCKCODE	REVENUE
1	DOT	206245.48

3. Identify products that have been sold in exactly 10 or more different invoices.

```
148
149
150
151  -- 3 Identify products that have been sold in exactly 10 or more different invoices
152
153  select
154      stockcode,
155      count(distinct invoiceno) as transaction
156      from retail
157      group by stockcode
158      having count(distinct invoiceno) >=10;
159
```

Results | Chart

	STOCKCODE	TRANSACTION
1	85123A	2246
2	71053	342
3	84406B	293
4	84029E	438
5	21730	139
6	22633	487

4. Count how many times each product has been sold and list those that have been purchased more than 5 times.

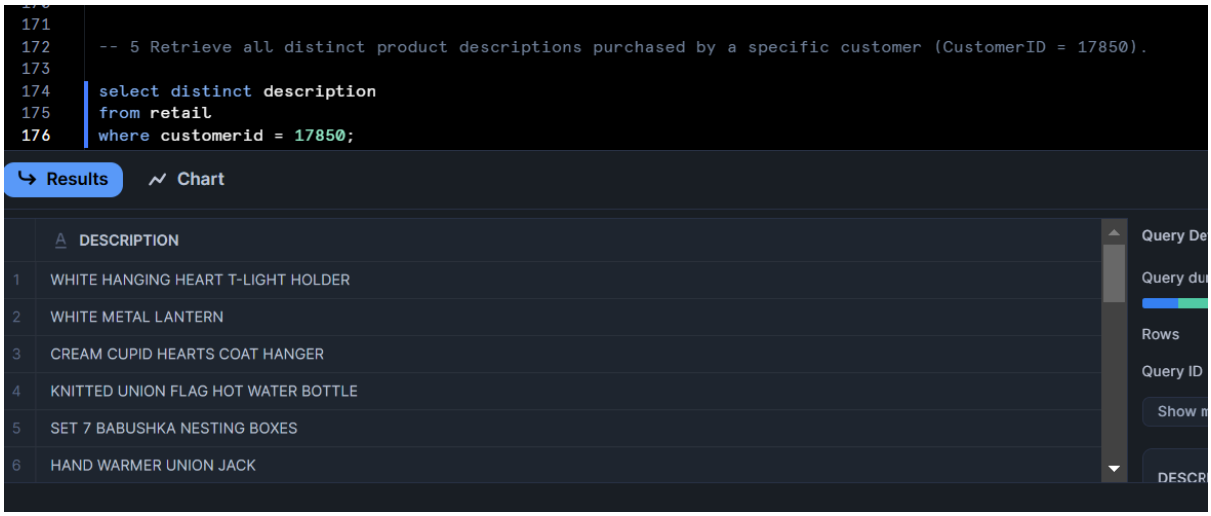
```
160
161  -- 4 Count how many times each product has been sold and list those that have been purchased more than 5 times.
162
163  select
164      stockcode,
165      count(distinct invoiceno) as transaction
166      from retail
167      group by stockcode
168      having count(distinct invoiceno) >5;
169
170
171
172  -- 5 Retrieve all distinct product descriptions purchased by a specific customer (CustomerID = 17850).
```

Results | Chart

	STOCKCODE	TRANSACTION
1	85123A	2246
2	71053	342
3	84406B	293
4	84029E	438
5	21730	139
6	22633	487

Query Details
Query duration
Rows
Query ID 01baa6fc-32
Show more

5. Retrieve all distinct product descriptions purchased by a specific customer (CustomerID = 17850).



4. Time bas analysis

1. Find the total revenue generated per month.

```sql
-- 1. Find the total revenue generated per month.

SELECT
    YEAR(InvoiceDate) AS InvoiceYear,
    MONTH(InvoiceDate) AS InvoiceMonth,
    SUM(Quantity * UnitPrice) AS Revenue
FROM retail
GROUP BY InvoiceYear, InvoiceMonth;
```

Results | Chart

	INVOICEYEAR	INVOICEMONTH	REVENUE
1	2010	12	748957.02
2	2011	1	560000.26
3	2011	2	498062.65
4	2011	3	683267.08
5	2011	4	493207.12
6	2011	6	691123.12

2. Identify the hour of the day when the highest number of transactions occurred.

```sql
-- 2 Identify the hour of the day when the highest number of transactions occurred.

SELECT
    HOUR(InvoiceDate) as InvoiceHour,
    COUNT(DISTINCT InvoiceNo) as Transactions
FROM retail
GROUP BY InvoiceHour
ORDER BY Transactions DESC
LIMIT 1;

-- 3. Count the number of invoices generated per day.
select
```

Results | Chart

	INVOICEHOUR	TRANSACTIONS
1	12	3962

3. Count the number of invoices generated per day.

```
200
201  -- 3. Count the number of invoices generated per day.
202  SELECT
203      DATE(InvoiceDate) AS InvDate,
204      COUNT(DISTINCT InvoiceNo) AS Transactions
205  FROM retail
206  GROUP BY InvDate;
207
```

Results | Chart

	INVDATE	TRANSACTIONS
1	2010-12-01	143
2	2010-12-02	167
3	2010-12-03	108
4	2010-12-05	95
5	2010-12-06	133
6	2010-12-07	111

4. Identify the date when the highest number of products were sold.

```
207
208  -- 4. Identify the date when the highest number of products were sold.
209
210  SELECT
211      DATE(InvoiceDate) as InvDate,
212      SUM(Quantity) as total_quantity
213  FROM retail
214  GROUP BY InvDate
215  ORDER BY total_quantity DESC
216  LIMIT 1;
217
```

Results | Chart

	INVDATE	TOTAL_QUANTITY
1	2011-10-05	46161

5. Find the number of transactions that happened before 12 PM vs. after 12 PM.

```
218
219
220  --5. Find the number of transactions that happened before 12 PM vs. after 12 PM.
221
222  SELECT
223      'Before 12 PM' AS Period,
224      COUNT(DISTINCT InvoiceNo) as Transactions
225  FROM retail
226  WHERE HOUR(InvoiceDate) < 12
227  UNION
228  SELECT
229      'After 12 PM' AS Period,
230      COUNT(DISTINCT InvoiceNo) as Transactions
231  FROM retail
232  WHERE HOUR(InvoiceDate) >= 12;
233
```

Results | Chart

	PERIOD	TRANSACTIONS
1	Before 12 PM	8627
2	After 12 PM	17274