# Data Analysis and Management Framework for E-commerce

This document provides a comprehensive analysis of sales and customer data, starts with data accuracy check's, Outline Requirements, Data Modeling Steps. Also Exploring key metrics such as transaction volumes, sales amounts, and purchasing trends across different demographics and regions, while also identifying actionable insights for enhancing customer engagement and driving business growth.

## I. Verify the Accuracy, Completeness, and Reliability of Source Data:

**Steps to Verify Data:**

- **Cross-Referencing:**

Compared the data against known reliable values to validate accuracy and completeness. Findings are some of the column's values in Customer table contains Special characters.

```
df.filter(regexp_extract(col('First'),"[^a-zA-Z]",0)!='').display()
```

Table

| | Customer_ID | First | Last | Age | Country |
|---|---|---|---|---|---|
| 1 | 6 | N!cole | Jones | 33 | USA |
| 2 | 14 | N!cole | Lara | 77 | UK |
| 3 | 109 | R0bert | Moore | 40 | UK |
| 4 | 118 | R0bert | Shepherd | 28 | UK |
| 5 | 162 | N!cole | Bennett | 51 | USA |
| 6 | 171 | L@rry | Cole | 50 | USA |
| 7 | 198 | R0bert | Bryan | 49 | UK |
| 8 | 211 | Al1cia | Thompson | 38 | USA |
| 9 | 214 | N!cole | Mcintyre | 18 | UK |
| 10 | 236 | Al1cia | Jensen | 19 | USA |

10 rows | 0.16 seconds runtime

**Validation Rules:** Defined specific validation rules based on the requirements (e.g., acceptable value, data type checks).

- **Data Profiling:**

I have used in-build data profiling feature to analyze the source data for anomalies, missing values, duplicates, and inconsistencies.

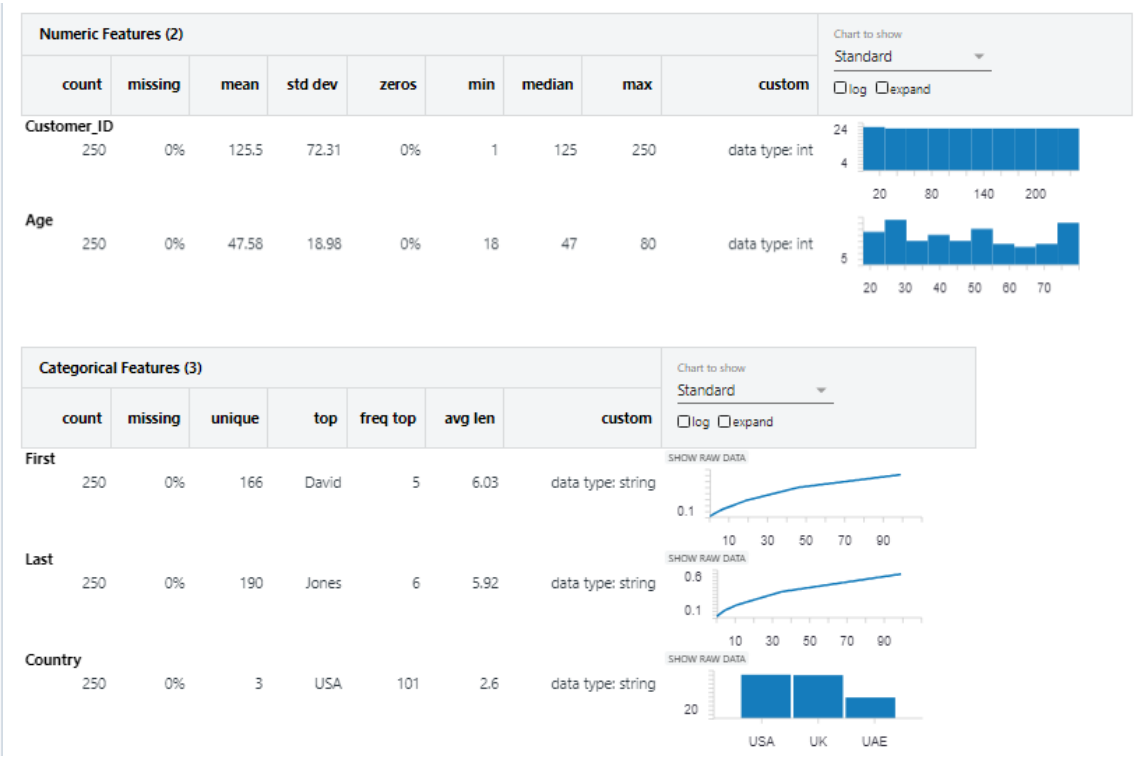



Fig: Data profiling for Customer table

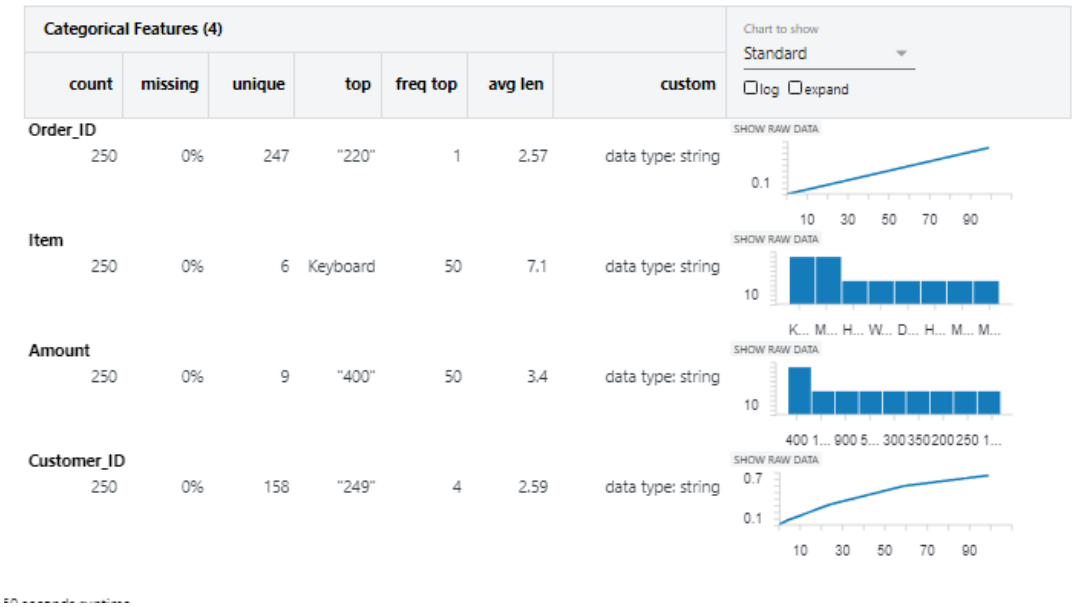



Fig: Data profiling for Order table

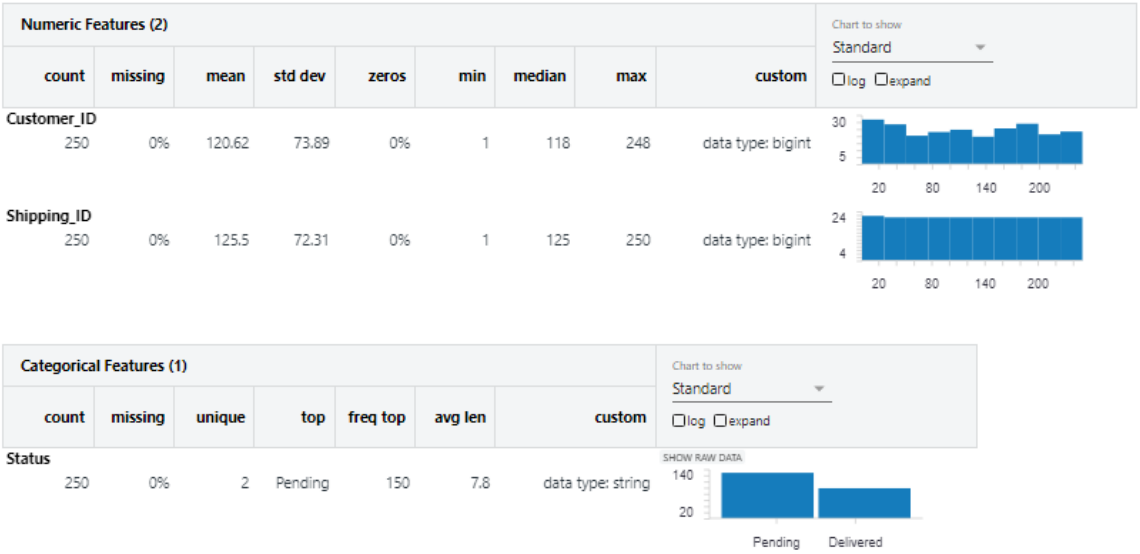



Fig: Data profiling for Shipping table

- **Verified Checklist:**

**Completeness**: Confirmed that all required fields are populated and check for missing records.

**Consistency**: Validated that data formats are uniform across datasets (e.g., date formats, naming conventions). But some of the column's values in Customer table contains Special characters.

**Timeliness**: Assess whether the data is up-to-date and relevant for current analysis needs.

## II. Requirements Specification:

### A. Data Components:

- Entities: Defined the main entities (e.g., Customers, Orders, Products).
- Attributes: Specify necessary attributes for each entity (e.g., Customer ID, Name, Age, Order Amount).
- Relationships: Outlined relationships between entities (e.g., one-to-many between Customers and Orders).

### B. Quality Requirements:

- Defined acceptable thresholds for data quality metrics (e.g., accuracy > 95%, completeness > 90%).

## III. Developed Data Models:

### High-Level Data Entities and Their Relationships

### 1. Entities Overview

1. Customer

- Attributes:
  - Customer_ID (Primary Key)
  - First_Name
  - Last_Name
  - Age
  - Country

2. Order

- Attributes:
  - Order_ID (Primary Key)
  - Customer_ID (Foreign Key)
  - Amount
  - Status

3.Shipping

- Attributes:
  - Shipping_ID (Primary Key)
  - Customer _ID (Foreign Key)
  - Status

**2. Relationships Between Entities**

- Customer to Order:
  - Type: One-to-Many
  - Description: A customer can place multiple orders, but each order is associated with only one customer.
  - Relation:
    - Customer_ID in the Order entity references Customer_ID in the Customer entity.
- Customer to Shipping:
  - Type: One-to-Many
  - Description: A customer can have multiple shipping records associated with their orders.
  - Relation:
    - Customer_ID in the Shipping entity references Customer_ID in the Customer entity.
- Order to Shipping:
  - Type: One-to-One

- Description: Each order has a corresponding shipping record, indicating the delivery status and details.
- Relation:
  - Order_ID in the Shipping entity references Order_ID in the Order entity.

Physical Model:

| Table | Column | Data Type | Constraints |
|---|---|---|---|
| Customer | Customer_ID | Int | Primary Key |
| Customers | First_Name | VARCHAR(50) | NOT NULL |
| Customers | Last_Name | VARCHAR(50) | NOT NULL |
| Customers | Age | int | CHECK (Age >= 0) |
| Customers | Country | VARCHAR(50) | |
| Orders | Order_ID | INT | Primary Key |
| Orders | Customer_ID | INT | Foreign Key references Customers.Customer_ID, NOT NULL |
| Orders | Amount | DECIMAL(10, 2) | |
| Orders | Item | VARCHAR(50) | ('Pending', 'Shipped', 'Delivered', 'Cancelled') |
| Shipping | Shipping_ID | INT | Primary Key |

| | | | |
|---|---|---|---|
| Shipping | Customer_ID | INT | Foreign Key references Customers.Customer_ID, NOT NULL |
| Shipping | Status | VARCHAR(50) | ('Pending', 'Shipped', 'Delivered') |

## IV. Technical Specifications:

### 1. Data Sources and Ingestion:

**The data pipeline will need to ingest data from the following sources:**

**Customer Data:**

Source: CRM Database (e.g, MySQL)( 'Customer.xls')

Key Columns: Customer_ID, First_Name, Last_Name, Country

**Order Data:**

Source: Order Management System(‘'Order.csv')

Key Columns: Order_ID, Customer_ID, Total_Amount, Status

**Shipping Data:**

Source: Shipping/Delivery Management API('Shipping.json')

Key Columns: Shipping_ID, Customer_ID, Status

### 2. Data Extraction

**we will extract data from the following sources:**

**Customer and Order Data:** Perform SQL queries to retrieve all necessary fields. The data should be extracted periodically (daily or as necessary) to capture new or updated records.

**Shipping Data:** Pull real-time shipping updates through API calls to the external shipping system and store it in a staging table.

**Data Extraction Strategy:**

Use incremental extraction to pull only new or updated records based on timestamps eg: (Order_Date, Shipping_Date,'_az_update_ts','_az_insert_ts').

Make API calls at regular intervals to ensure the most recent shipment statuses are captured.

**3. Data Transformation**

The transformation step involves:

**Data Cleansing:**

Dropping  duplicates (based on Customer_ID or Order_ID).

Handled missing or null values (e.g., shipping status, order amounts).

Handled special characterises (e.g., Frist name status, last name).

Standardized text fields

**Joining Tables:**

Customer with Shipping: Join on Customer_ID to obtain the shipping status for each order.

(Customer with Shipping)join with Orders: Join on Customer_ID to get each customer's order details.

**Derived Columns and Calculations:**

1. Total Amount Spent and Country for the Pending Delivery Status for Each Country

Objective: Calculated the total monetary value of transactions for orders that are still in a "Pending" delivery status and aggregate this by each country.

Approach:

Filtered the dataset to include only records with the "Pending" delivery status. Grouped data by Country to summarize the Total_Amount for pending orders.Output a result showing each country and the corresponding total pending amount.

2. Total Number of Transactions, Total Quantity Sold, and Total Amount Spent for Each Customer, Along with Product Details

Objective: For each customer, need to calculate:

Total number of transactions they have made. Total quantity of products they have purchased (even though there is no explicit quantity column, we need to calculate it based on order data or product counts). Total monetary value of all their transactions. Include details about the products they purchased.

Approach:

Group data by Customer_ID and aggregating:

Count the number of orders per customer (total transactions). Calculate total Total_Amount spent by each customer.Determine the total quantity sold using derived logic if Quantity is absent.Use joins to bring in product details (such as Product_Name and Product_ID).The result would be a table of customers, with their transaction totals, amounts, and products purchased.

3. The Maximum Product Purchased for Each Country

Objective: Identified the product that has been purchased the most in each country, based on the total quantity or number of transactions.

Approach:

Grouped data by Country and Product_ID, and count how many times each product was purchased (or infer quantity).For each country, rank or order the products by the count of purchases or quantity sold.Extract the top-ranked (maximum purchased) product for each country.Output the product name and the total count/quantity for that country.

4. The Most Purchased Product Based on Age Category (Less than 30 and Above 30)

Objective: Determine which product is the most popular for two distinct age groups: customers aged less than 30 and those aged 30 and above.

Approach:

First, create a derived column categorizing customers into two age groups: <30 and >=30.Group the data by Age_Category and Product_ID, and count the number of times each product was purchased. For each age group, rank or order the products by their purchase count.Extract the top-ranked (most purchased) product for each age category.

5. The Country That Had the Minimum Transactions and Sales Amount

Objective: Identify the country that has the lowest number of transactions and the least amount of total sales.

Approach:

Group the data by Country, and aggregate:

Count the total number of transactions (orders) per country.Sum up the Total_Amount for each country to get the total sales amount.Rank the countries based on the number of transactions and total sales.Extract the country with the lowest transaction count and sales amount.

Transformation Logic:

Used PySpark for distributed processing.

Implemented transformations using DataFrame APIs such as groupBy, agg, join, and filter.

**4. Data Loading:**

The transformed data will be loaded into the following target tables in the data warehouse (e.g., Azure Storges account):

**Fact Tables:**

Orders_Fact(temp_view): Contains the final order data, including customer and product details, total amount spent, and shipping status.

**Dimension Tables:**

Customer_Dim(temp_view): Enriched customer data, including demographic details like age and country.

Shipping_Dim(temp_view):Contains shipping information, including status and tracking.

**Data Loading Strategy:**

Incremental Loading:

Only new or updated records should be loaded to avoid duplication.Batch or Stream Processing: Depending on the size and velocity of the data, implement batch loads (daily) or real-time stream loads (for APIs).

Indexes:

Create indexes on frequently queried columns (Customer_ID, Order_ID) to optimize query performance.

**5. Data Validation and Testing:**

To ensure data accuracy and reliability, implement the following validation checks:

Data Completeness:

Verify that all expected fields are populated (e.g., Customer_ID, Order_Date, Total_Amount).

Data Integrity:

Ensure that joins between tables (e.g., Customer_ID in Orders and Customers) result in correct and complete data.

Business Logic Validation:

Ensure that calculated fields (e.g., total sales, most purchased Item) are consistent with source data.

Unit Tests:

Create unit tests for key transformations to ensure they produce the expected output.

Validate those filters (e.g., age < 30) work correctly.

**6. Performance Optimization**

Partitioning:Partition data by Order_Date or Customer_ID for efficient query processing.

Caching:Cache intermediate DataFrames in memory to optimize repeated calculations (e.g., total amount per customer).

Indexing:Index keys such as Customer_ID, and Order_ID in the target warehouse to speed up lookup queries.

## V. Communicating Findings and Insights

**Visualization Techniques:**

To communicate the insights effectively to stakeholders, you can leverage the following visualization techniques:

**Pie Chart: Total Amount Spent for Pending Deliveries by Country**

**Purpose:**

- The goal of this pie chart is to show the proportion of total pending amount that each country contributes. This allows stakeholders to quickly see which countries have the largest or smallest pending deliveries.

**Key Insights:**

1. **Country Contributions:**
    - The size of each slice represents the total pending delivery amount for that country as a proportion of the overall pending delivery total.
    - Larger slices indicate countries with more pending orders in terms of monetary value.
2. **Visual Representation:**

- Color Coding: Each country is represented by a different color. This makes it easy to distinguish between countries.

- Labels: The total pending amount (or percentage share) can be displayed directly on each slice, providing quick access to the exact values.

**Example:**

Let's say we have pending delivery data for three countries:

- **UK**: $ 1,36,300 (53.33% of total pending amount)
- **UAE**: $ 53,800 (21.3% of total pending amount)
- **USA**: $65,500 (25.63% of total pending amount)

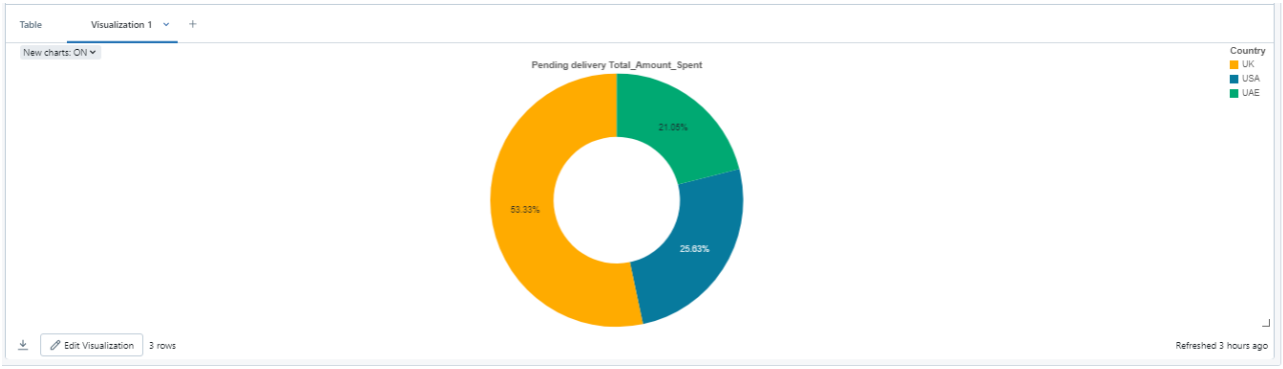
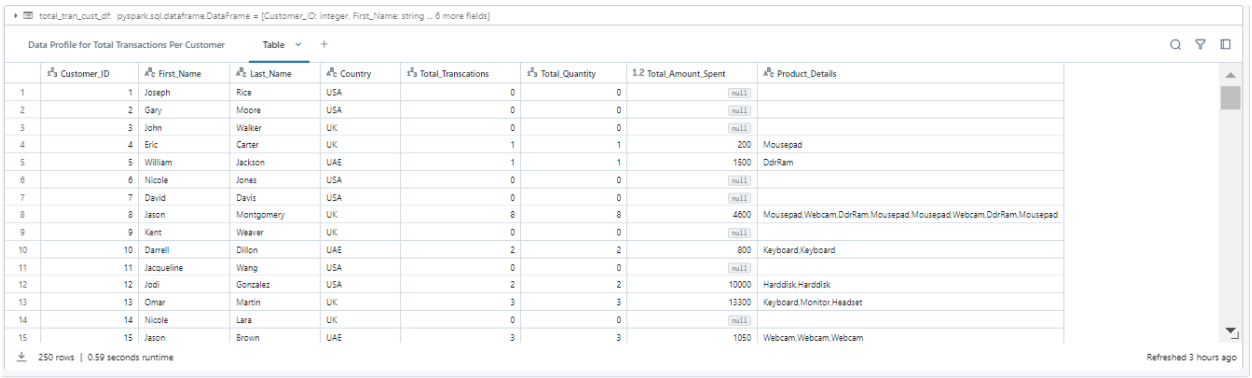



Fig: the total number of transactions, total quantity sold, and total amount spent for each customer, along with the product details.




| | Customer_ID | First_Name | Last_Name | Country | Total_Transcations | Total_Quantity | Total_Amount_Spent | Product_Details |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | Joseph | Rice | USA | 0 | 0 | null | |
| 2 | 2 | Gary | Moore | USA | 0 | 0 | null | |
| 3 | 3 | John | Walker | UK | 0 | 0 | null | |
| 4 | 4 | Eric | Carter | UK | 1 | 1 | 200 | Mousepad |
| 5 | 5 | William | Jackson | UAE | 1 | 1 | 1500 | DdrRam |
| 6 | 6 | Nicole | Jones | USA | 0 | 0 | null | |
| 7 | 7 | David | Davis | USA | 0 | 0 | null | |
| 8 | 8 | Jason | Montgomery | UK | 8 | 8 | 4600 | Mousepad,Webcam,DdrRam,Mousepad,Mousepad,Webcam,DdrRam,Mousepad |
| 9 | 9 | Kent | Weaver | UK | 0 | 0 | null | |
| 10 | 10 | Darrell | Dillon | UAE | 2 | 2 | 800 | Keyboard,Keyboard |
| 11 | 11 | Jacqueline | Wang | USA | 0 | 0 | null | |
| 12 | 12 | Jodi | Gonzalez | USA | 2 | 2 | 10000 | Harddisk,Harddisk |
| 13 | 13 | Omar | Martin | UK | 3 | 3 | 13300 | Keyboard,Monitor,Headset |
| 14 | 14 | Nicole | Lara | UK | 0 | 0 | null | |
| 15 | 15 | Jason | Brown | UAE | 3 | 3 | 1050 | Webcam,Webcam,Webcam |

**Columns Explained:**

1. **Customer_ID**:
    - Unique identifier for each customer.
    - This helps in identifying individual customers and their associated data.

2. **Total_ Transactions( Number of Transactions)**:
    - Represents the **total number of orders or transactions** the customer has made.
    - This is an aggregation of all purchases made by that customer.
3. **Total_ Quantity (Total Quantity Sold)**:
    - The total number of units/products bought by the customer.
    - If quantity information isn't directly available, it may be inferred from other columns (e.g., if a customer ordered the same product multiple times).
4. **Total_Amount _Spent**:
    - The total monetary value of all transactions made by the customer.
    - It is the sum of all purchases across transactions for that customer.
5. **Product_Details**:
    - This column lists all the unique products purchased by the customer along with the quantity of each product.

**Bar Chart: Maximum Product Purchased by Country**

**Purpose:**

- The bar chart visually represents the product that has been purchased the most in terms of quantity or value in each country.
- Each bar corresponds to a country and displays the product that had the highest total purchases (either by units or revenue).

**Chart Setup:**

1. **X-axis (Country)**:
    - Represents different countries.
    - Each country will have one bar showing the most popular product.
2. **Y-axis (Amount)**:
    - This axis will represent the **total amount spent** for the most purchased product in that country.
    - **Total Amount Spent**: If you want to show which product generated the most revenue.
3. **Bars (Products)**:
    - Each bar will show the **maximum purchased product** for that country.
    - The bar height reflects the total number of units sold or the total amount spent on that product.

**Area Chart: Most Purchased Product by Age Category**

**Purpose:**

- The area chart visualizes how product purchases vary across the two age groups: **less than 30** and **30 or older**.
- It shows the number of units sold (or total amount spent) for each product over these two age categories, allowing stakeholders to compare product preferences between the two groups.

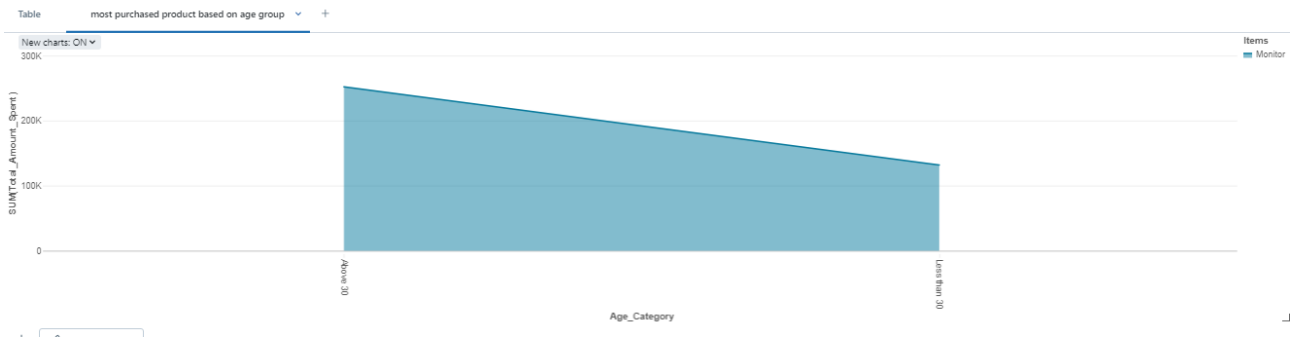



**Chart Setup:**

1. **X-axis (Products)**:
    - The horizontal axis will represent **different products**.
    - Each product (e.g., "Keyboard," "Mouse," "Monitor") will be placed along the X-axis.
2. **Y-axis (Quantity or Amount)**:
    - The vertical axis will represent either the **total quantity sold** or the **total amount spent**.
    - This reflects the total sales (units or amount) for each product by age category.
3. **Areas (Age Categories)**:
    - The chart will have two overlapping shaded areas:
        - **Age < 30**: One area represents purchases made by customers under 30 years old.
        - **Age ≥ 30**: Another area represents purchases made by customers 30 years old or older.
    - The size of each area indicates how much that age group purchased of a particular product.
4. **Color Coding**:
    - Use two distinct colors to represent the two age categories. For example:

- **Age < 30**: Light blue.
- **Age ≥ 30**: Dark blue.

- The areas may overlap for certain products, visually indicating where preferences align or differ between age groups.

**Table: Country with Minimum Transactions and Sales Amount**

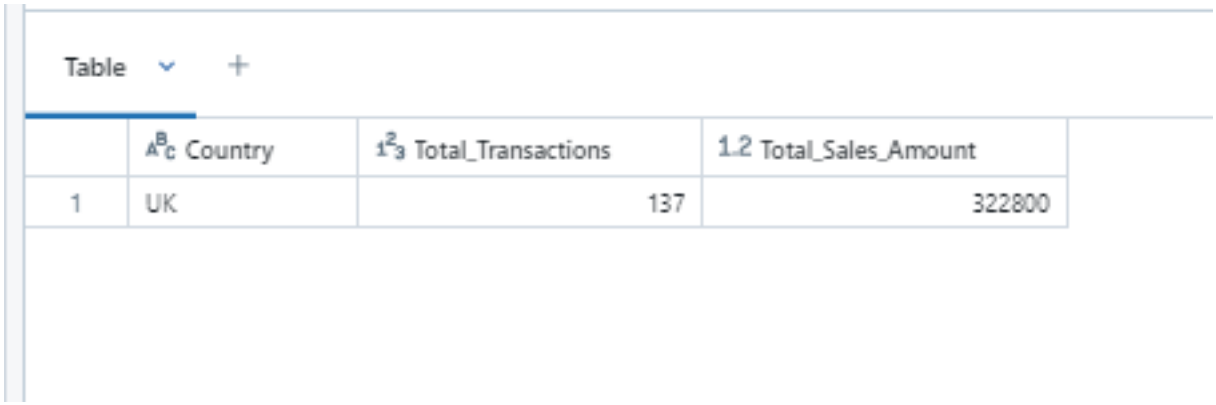


Table +

| | Country | Total_Transactions | Total_Sales_Amount |
|---|---|---|---|
| 1 | UK | 137 | 322800 |

**Columns Explained:**

1. **Country**:
   - The name of the country for which the minimum transactions and sales amount are recorded.
   - This allows for easy identification of the region in question.
2. **Total Transactions**:
   - This column shows the **total number of transactions** made in that country.
   - It helps understand the level of activity or engagement with customers in that region.
3. **Total Sales Amount**:
   - The monetary value of all transactions made in that country.
   - This reflects the total revenue generated from sales in that country.

we have aggregated data from your datasets and determined that **UK** has the fewest transactions and the lowest sales amount, the table would look as follows:

- **Country**: UK
- **Total Transactions**: 137
- **Total Sales Amount**: $322800 (indicating the total revenue generated from those purchases).

In summary, our analysis of sales and customer data reveals important trends and areas for improvement, helping us create targeted strategies to boost sales and better engage with our customers.