E-Commerce Database Management System Project

Group 8

Table of Contents

**Load Necessary Libraries**

library(DBI)  
library(readr)  
library(RSQLite)  
library(dplyr)  
library(stringr)

# Introduction

This project simulates a real-world e-commerce data environment, including all stages of data management, from database configuration to data analysis and reporting. The database is created and managed with SQLite while GitHub Actions are used to automate processes enhancing workflow efficiency. In addition, Quarto is used in conjunction with R to conduct comprehensive data analysis and generate reports. Through this approach, technical skills and competencies necessary for effective data management are examined in-depth, mirroring the complexities and dynamics encountered in e-commerce.

# Database Design and Implementation

## Entity Relationship Diagram

**Figure 1** shows the E-R diagram that simulates a real-world e-commerce data ecosystem, capturing the detailed relationships between entities and attributes essential for facilitating online transactions. In addition, it provides a comprehensive view of the e-commerce system, which serves as a platform for users to browse products, make purchases, and securely complete their payments.

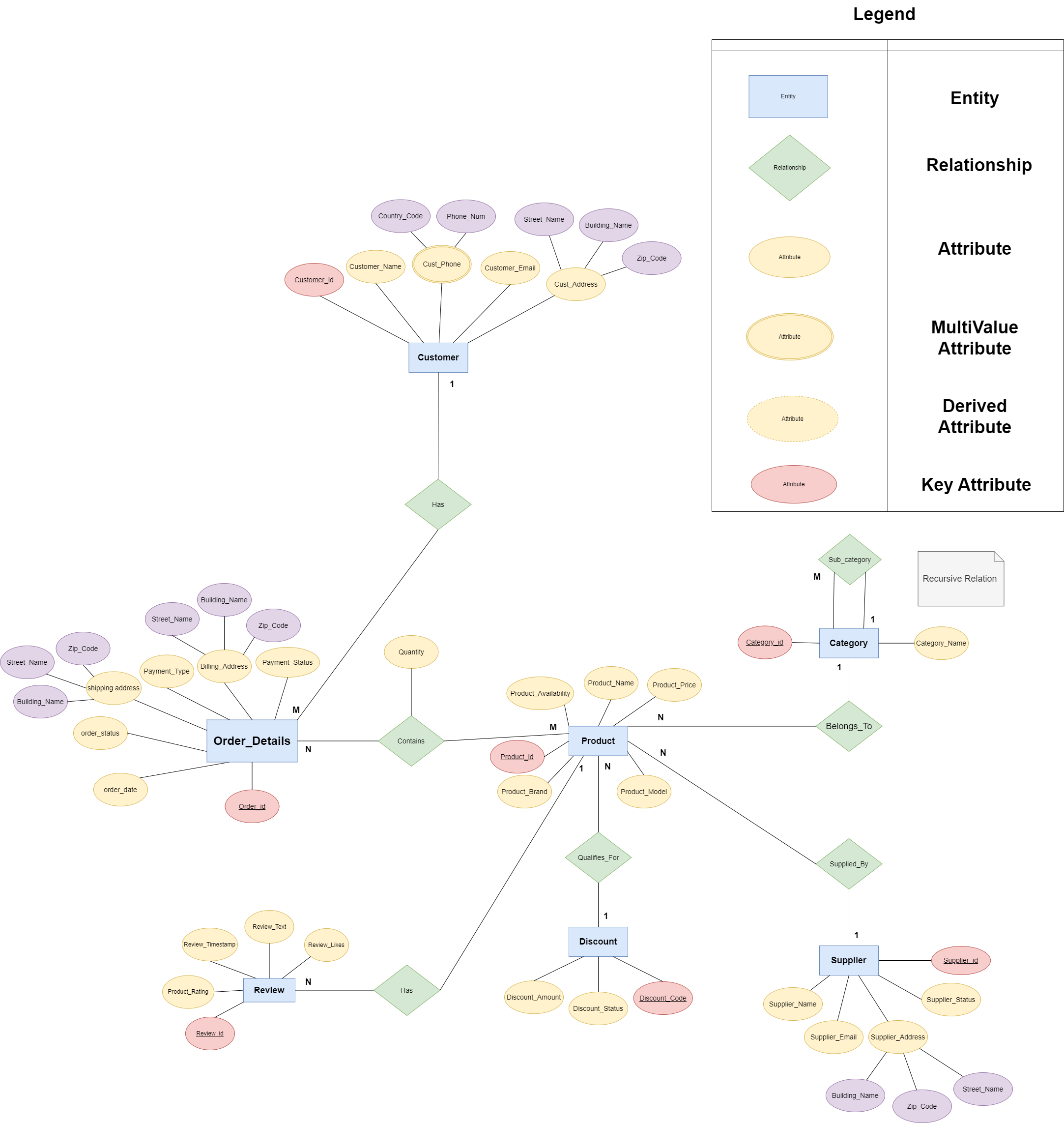


Figure 1: E-Commerce Entity Relationship Diagram

### Assumptions

* The company only distributes products within the United Kingdom (UK).
* The Currency used is Pound Sterling (GBP).
* Attributes formats will be aligned with UK standard formats such as date , addresses , names …etc

### Entities and Attributes

This section describes and illustrates the entities in the above ERD and their respective attributes.

* Customer

    Shows us the users who have placed orders in the past. Attributes include name , email, phone number and address of the customer.

* Supplier

    Vendors who supply products.This entity contains attributes of names, addresses, emails and status of whether the supplier is currently supplying product or not (Active/Inactive).

* Product

    Describes all products in the stock and available for sale. Attributes include name, price, and availability of the products.

* Order Details

    Emphasises all details related to placed orders. Some of its attributes includes billing and shipping addresses, type and status of payment, date and status of the order.

* Category and Sub-Category

    Category is the broad classification of products that share common features or are intended for a similar purpose. A sub-category is a more specific grouping of products within a category based on finer distinctions or attributes.

* Product Discounts

    The voucher number or offer code to be applied to eligible products. The amount of discount it offers as well as the status of the discount are the attributes included.

* Reviews

    Contains Written comments and rating of product bought by customers, the likes recieved as well as the time stamp of when the review was written.

## Design Considerations

### Absence of an Order Entity

The model intentionally skips direct order management. Instead, it focuses on product management and customer interactions through reviews and payment methods.

Order Entity not considered in this ER design in order to follow best practices by not having to include orderId as part of product table which might affect the overall performance of DB retrieval.

## Relationships and Cardinalities

### Order Detail Contains Products

This relation as can be seen in **Figure 2** indicates that each order detail contains multiple products that have been bought by the customer.

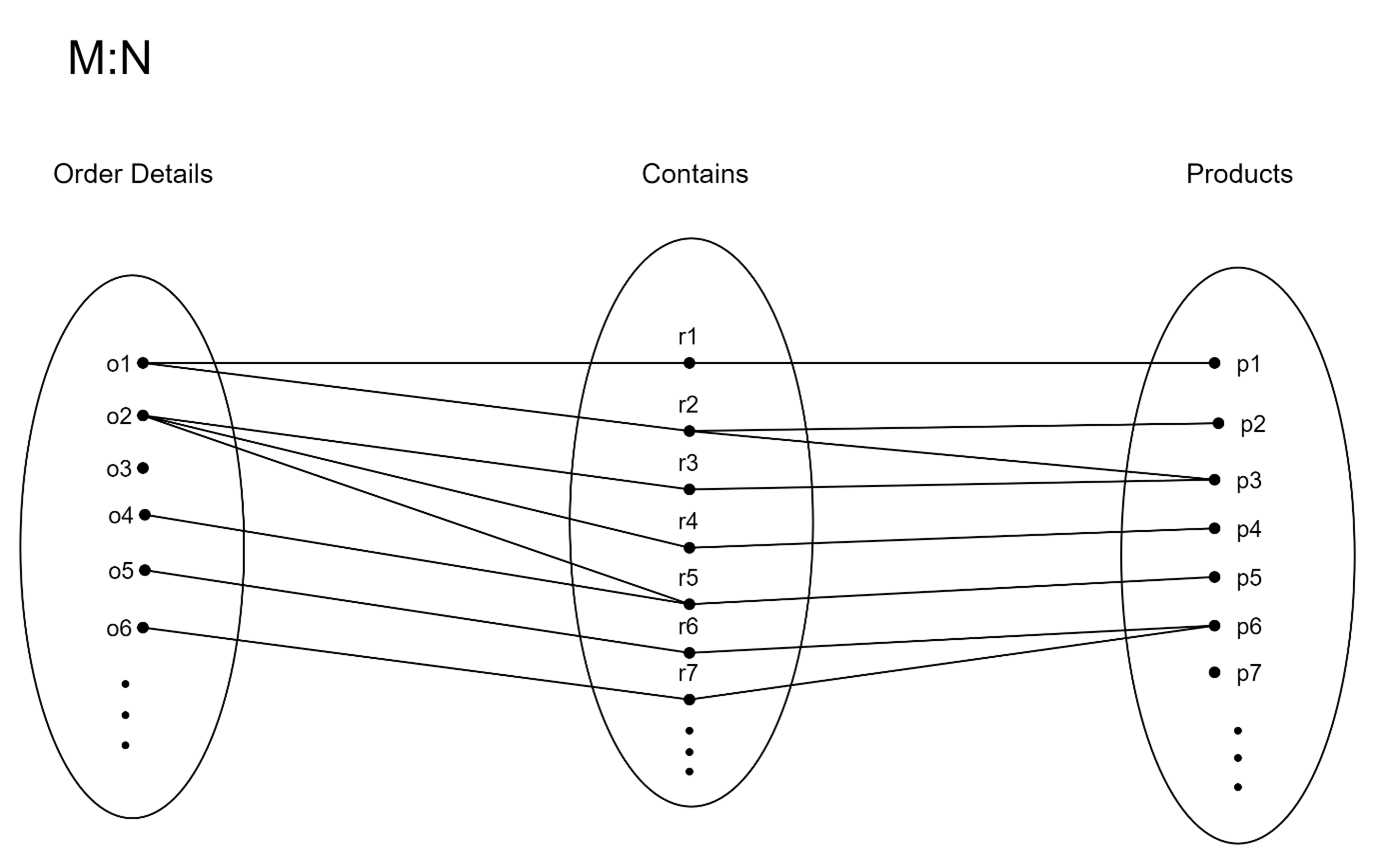


Figure 2: Order Details Contains Products

**Associative Attribute** (Quantity: The number of units of the product ordered in this line item.)

### Customer Has Order Details

This relation will be created when customers orders one or more products. They will be linked with a particular order detail indicating the order status throughout the process. One customer can be associated with multiple order details at any given time as illustrated in **Figure 3**. Moreover, it aids in tracking an order’s life cycle, allowing for updates, customer notifications, and management of the order fulfillment process.

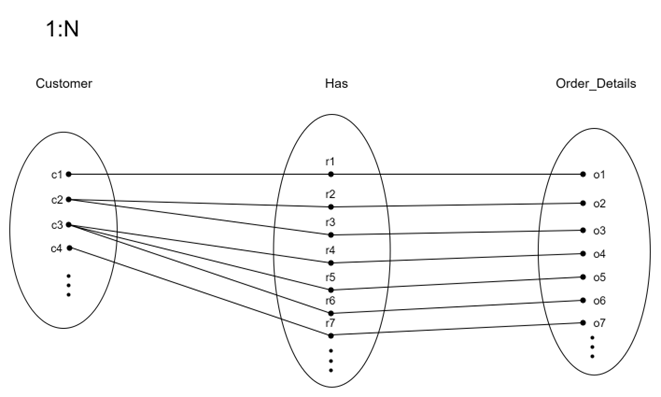


Figure 3: Customer Has Order Details

### Product Belongs to Category

**Figure 4** specify that each product is classified under a specific Category where products can belong to only one category. This enables customers to browse products by category and helps retailers manage product listings more efficiently.

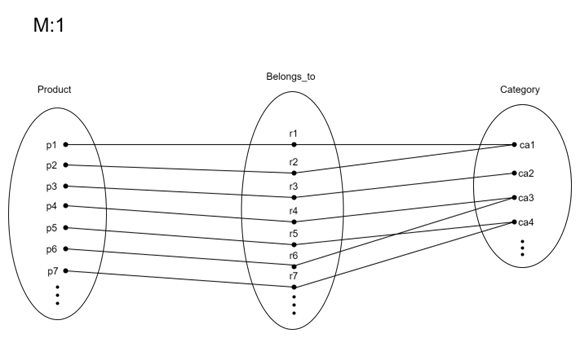


Figure 4: Product Belong to Category

### Category Self-Reference Relation

As reflected in **Figure 5**, category can have multiple subcategories. For example, “Home Appliances” is the parent category containing subcategories like Washing Machines, Food Processors, Dishwashers etc.

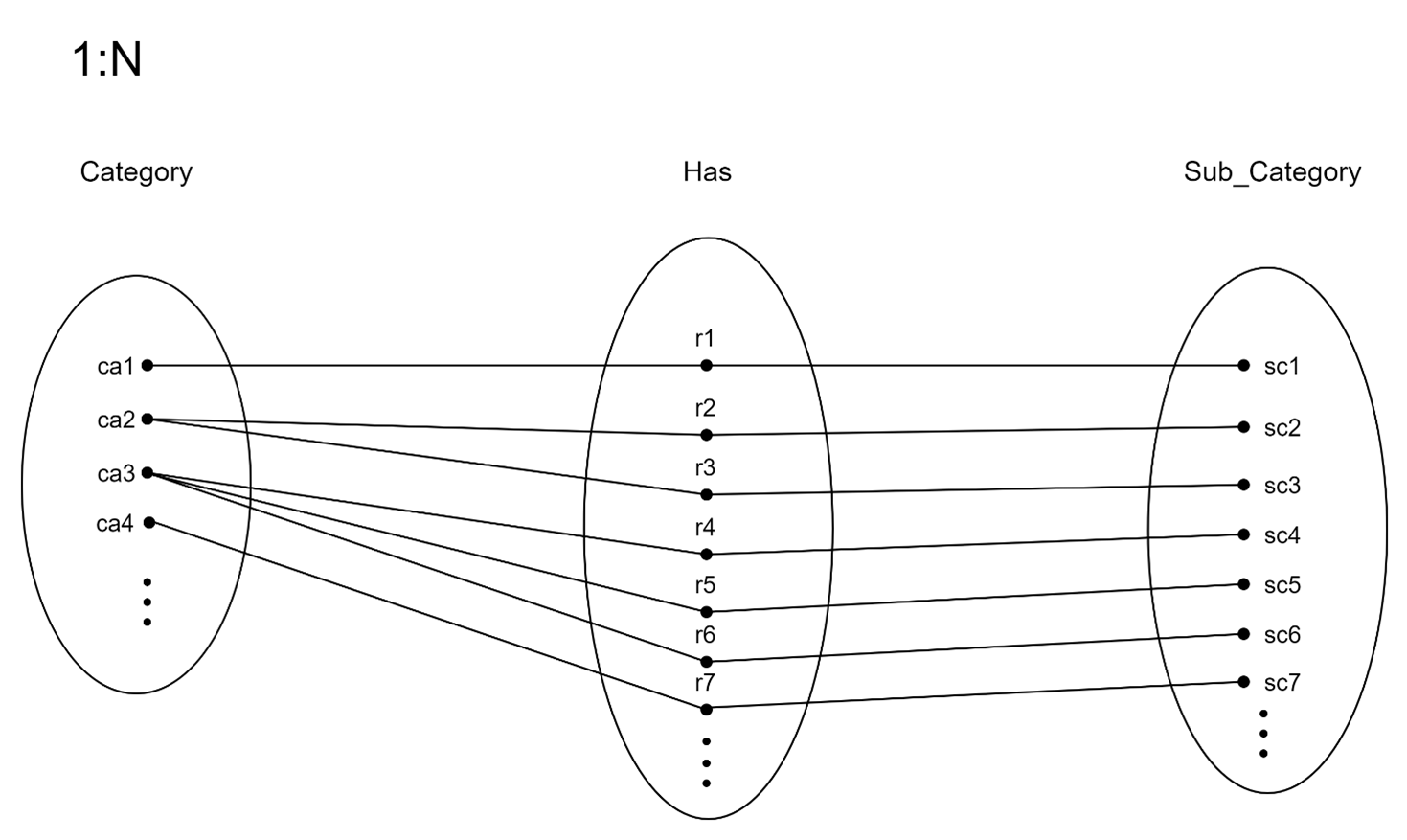


Figure 5: Category Has Sub-Categories

### Product Supplied By Supplier

This relation is important for inventory management, reflecting the real-world scenario where a single supplier might provide various products, but each product is supplied by only one specific supplier **Figure 6**. The relation helps track inventory sources, manage supplier relationships, and ensure product availability.

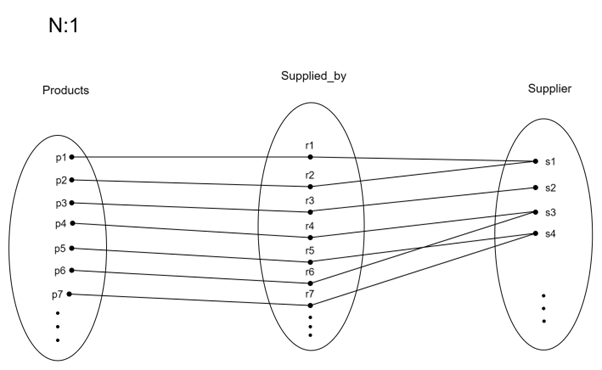


Figure 6: Products Supplied By Supplier

### Product Qualifies For Discount

This relation signifies that a product can be eligible for a promotional discount. In this context and for simplicity the relation representing one discount code or voucher that is valid to apply on multiple eligible products **Figure 7**.

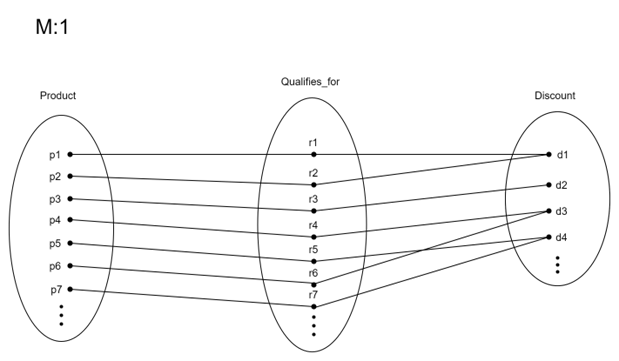


Figure 7: Product Qualifies For Discount

### Product Has Reviews

A product’s reviews are provided by Customers reflecting feedback or evaluation for a specific product or service. One product can have multiple reviews over time as shown in **Figure 8**.

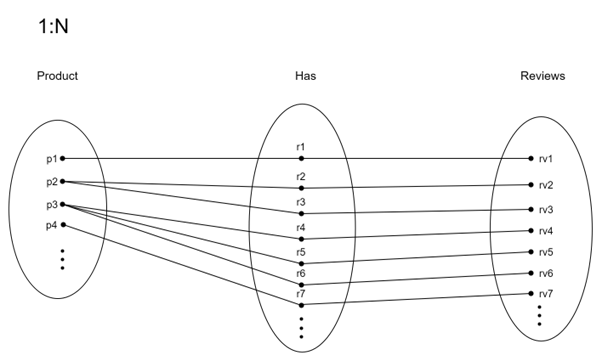
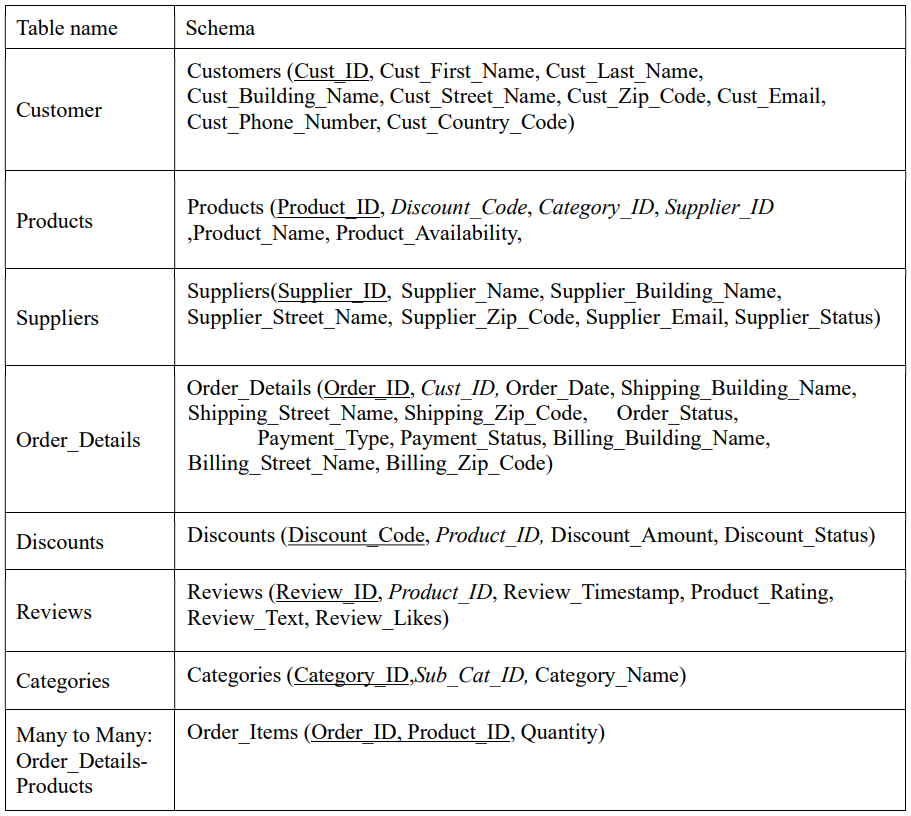


Figure 8: Product Has Reviews

## Logical Schema

The table below describes the logical schema of our E-commerce database. Italic attributes signify foreign keys.



## Physical Schema Creation

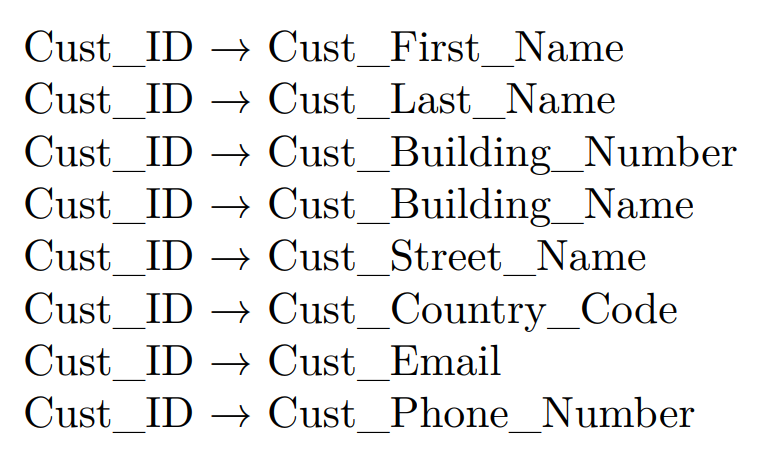
#Creating the db schema in SQL  
con <- dbConnect(RSQLite::SQLite(), "ecommerce.db")  
sql\_file <- readLines("dbScript.sql")  
for (sql\_command in sql\_file) {  
 if (sql\_command!=""){  
 print(sql\_command)  
 dbExecute(con,sql\_command)  
 print("-------------DONE---------")  
 }  
}

## Normalisation

The First Normal Form (1NF) mandates atomic values and unique rows via a primary key without duplicate columns. The tables were verified to meet these criteria. Second Normal Form (2NF) is achieved when a table in 1NF has all non-key attributes fully dependent on the primary key, thus avoiding partial dependencies on composite keys which has also been achieved. Finally, the team conducted thorough checks to ensure all tables comply with 3NF requirements, as detailed in subsequent sections.

### Customers

Looking at the table, we can identify the following dependencies:



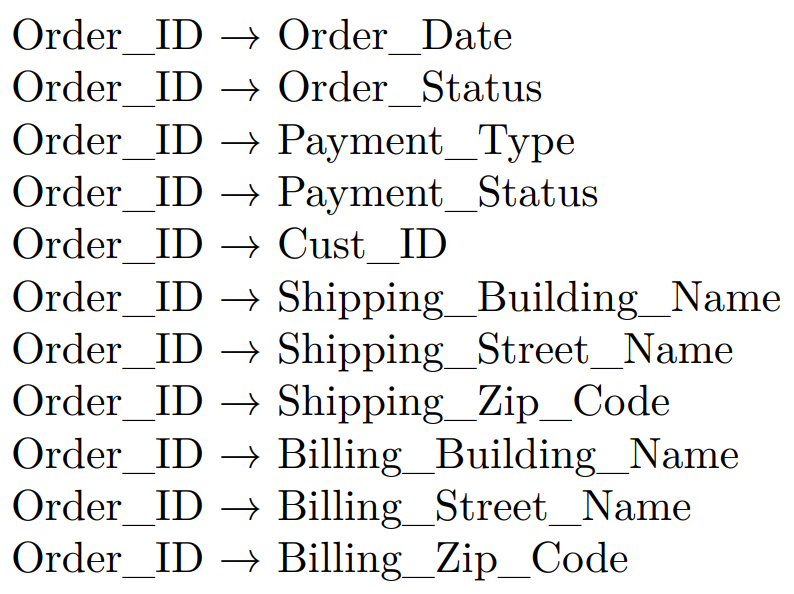
* Based on this analysis, the table appears to be in 3NF. Each non-key attribute depends only on the primary key (Cust\_ID), and there are no transitive dependencies.

### Order\_Items



Product\_ID and Order\_ID combined are representing composite key, and Quantity is functionally dependent on Product\_ID and Order\_ID. There are no transitive dependencies here.

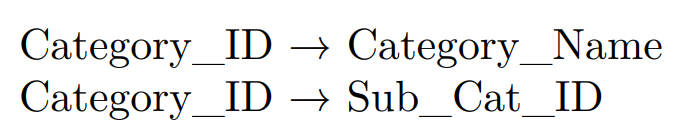
### Order\_Details



Given that Cust\_ID is a foreign key in this table and that there is a separate Customers table where Cust\_ID serves as the primary key, it suggests that Cust\_ID is not functionally dependent on any other attribute within this table. It is merely referencing the primary key of another table.

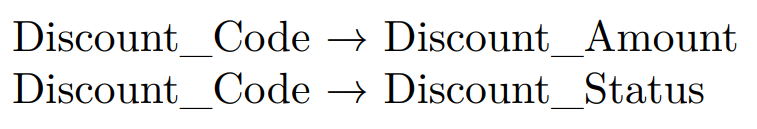
Therefore, the dependency involving Cust\_ID does not violate 3rd Normal Form (3NF). The table appears to be in 3NF as all other attributes are functionally dependent only on the primary key (Order\_ID)

### Product Category



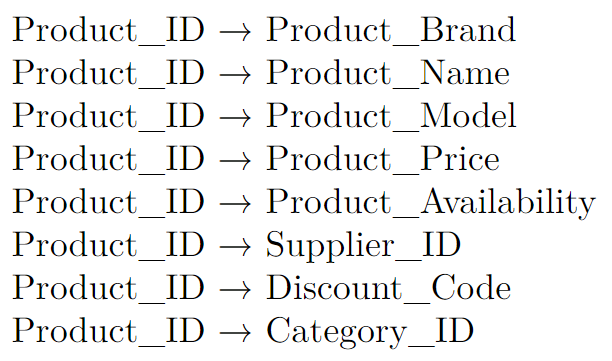
Category\_ID is the primary key and it uniquely identifies Category\_Name.

### Discount



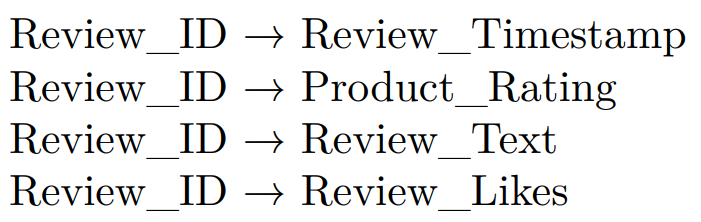
Analyzing the functional dependencies, it is found that Discount\_Code is the primary key and it uniquely identifies Discount\_Amount and Discount\_Status. Each Discount\_Code corresponds to a specific Discount\_Amount and Discount\_Status, ensuring that the table adheres to the principles of 3rd Normal Form (3NF). There are also no transitive dependencies or non-key attributes determining other attributes within the table.

### Products



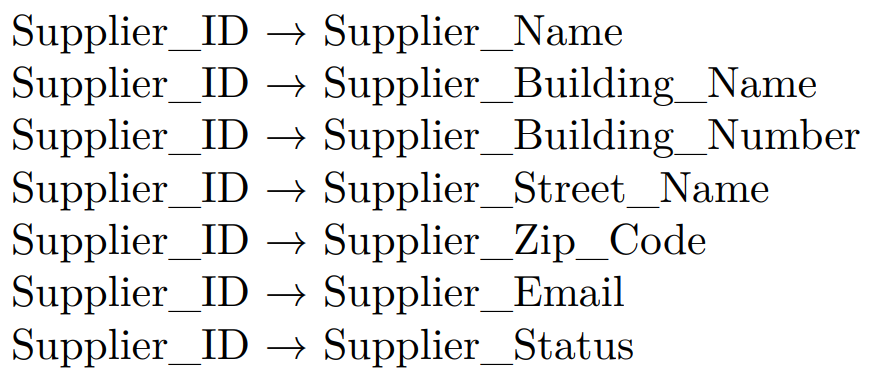
The table also contains foreign keys, namely Supplier\_ID, Discount\_Code, and Category\_ID. These foreign keys are merely referencing the primary key of their respective tables and are not functionally dependent on any other attribute within this table. Examining the non key attributes’ functional dependencies, Product\_ID emerges as the primary key, in which Product\_Name, Product\_Price and Product\_Availability appears to be functionally dependent on it.

### Reviews



In the Reviews table, Review\_ID acts as the primary key, ensuring uniqueness. Attributes like Review\_Timestamp, Product\_Rating, Review\_Text and Review\_Likes are dependent on Review\_ID.

### Suppliers



In the Suppliers table, Supplier\_ID acts as the primary key. Attributes such as Supplier\_Name, Supplier\_Building\_Name, Supplier\_Building\_Number, Supplier\_Street\_Name, Supplier\_Zip\_Code, Supplier\_Email and Supplier\_Status are dependent on Supplier\_ID.

Given these functional dependencies where each attribute seems to be functionally dependent on the primary key, with no non-key attributes determining other, all tables appears to adhere to the principles of 3rd Normal Form (3NF).

# Data Generation and Management

## Synthetic Data Generation

After the agreement on the schema mentioned in the previous section, the team started to generate synthetic data that to some extent, imitated realistic e-commerce as much as possible.

ChatGPT has been used as the main tool for this step as an alternative to Mockaroo, as the former produces more structural and logical data than the latter. **[Appendix 1,2]**. For tables consisting of foreign keys, R code has been used for assignment based of logic set by the team [**Appendix 3**](#appendix-3).

## Data Import and Quality Assurance

To enhance data quality for e-commerce analysis and validate the csv data generated, we use R to perform 4 main checks as shown in Figure 9 **[Refer data\_validate.R]**:

* Primary key integrity check
* Duplicate entry check
* Email id pattern check
* Phone number pattern check

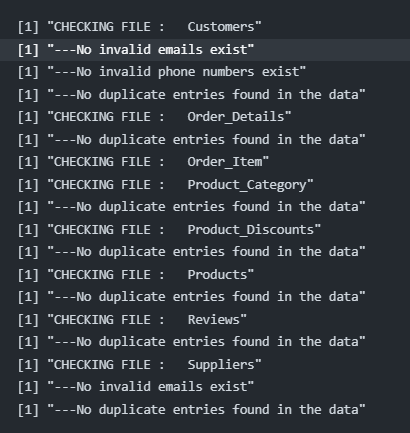


Figure 9: Data validation of CSV files

Once the data has gone through all the checks, the updated csv files will be used to import into SQL. Importing the data into SQL is also done in R **[Refer data\_update.R]**. This method focused on refining datasets for accurate analysis by identifying and eliminating invalid entries and duplicates, thereby maintaining the dataset’s uniqueness and reliability before inserting into Data Base.

# Data Pipeline Generation

To efficiently collaborate on this assignment, we have uploaded the project to Github. This was done so that the team could track changes, follow progress, collaborate on bug fixes etc. Moving the project to Github also allowed for implementation of version control and continuous integration.

## Github Repository Creation

The pipeline generation process consists of two phases. In the first phase, github repository has been created followed by integrating the posit cloud project. Additionally, all teams members have been added as collaborators.

## Automated Workflow

Second phase was the automating our workflow. Using Github Workflows, we were able to automate the process of validating data, updating data and data analysis. The workflow is triggered on pushing new data to the project. Once all 3 steps are completed successfully, the new files are committed and pushed to the project. Figures 10, 11, 12, 13 show how the db and analyses carried out is automatically updated after a workflow run

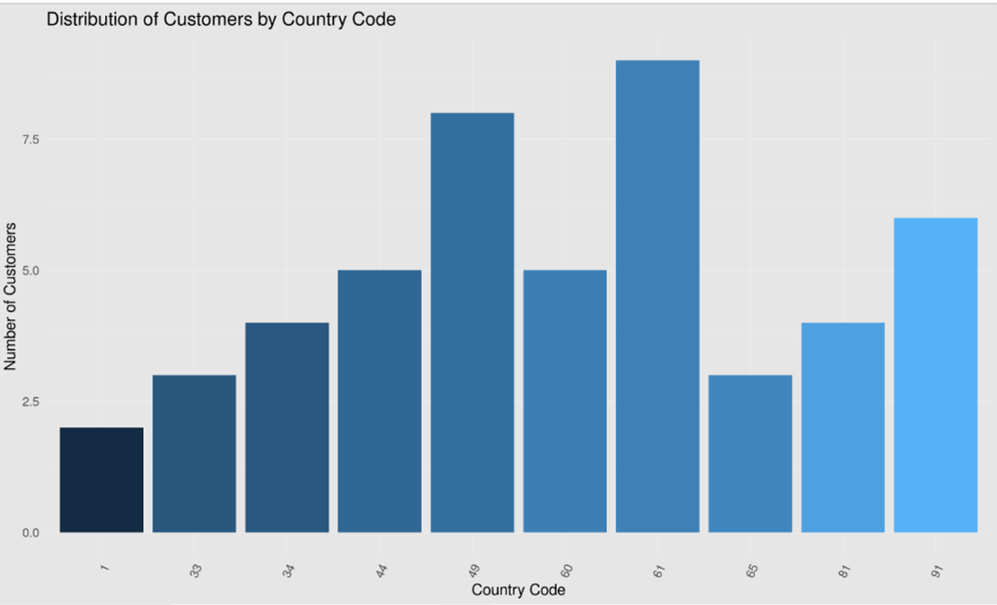


Figure 10: The histogram of Customer data for 50 records

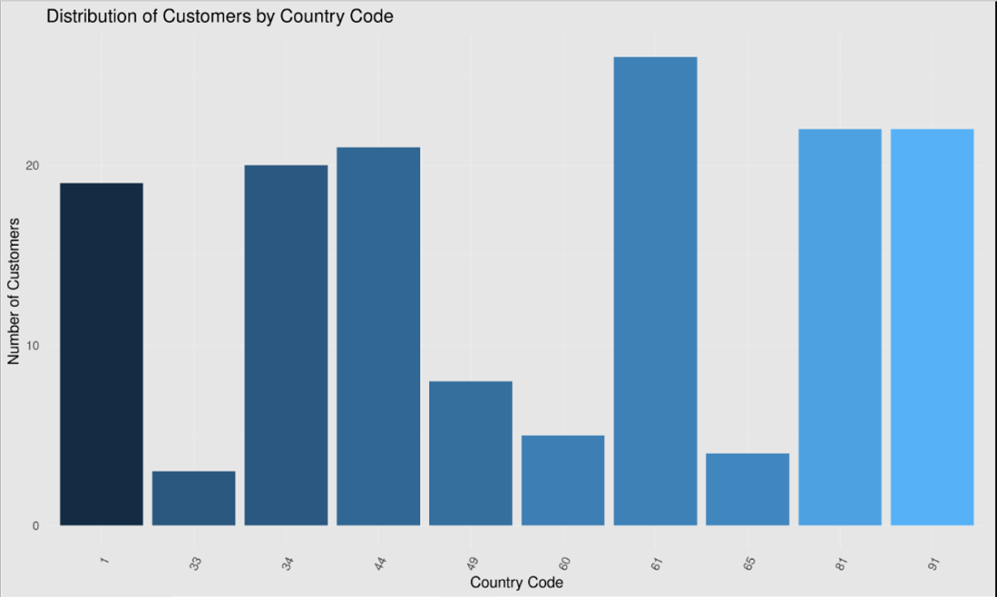


Figure 11: The histogram after workflow run and inserted additional 100 records

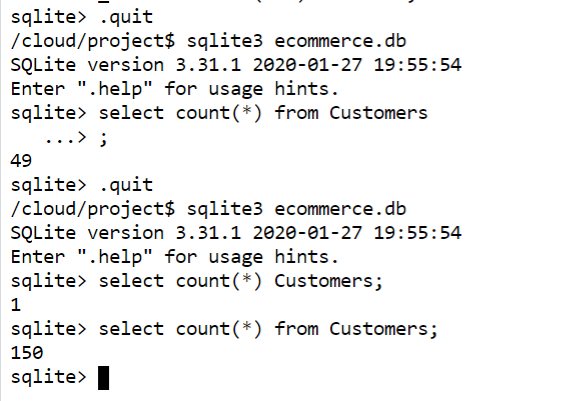


Figure 12: Record count after data update using workflow

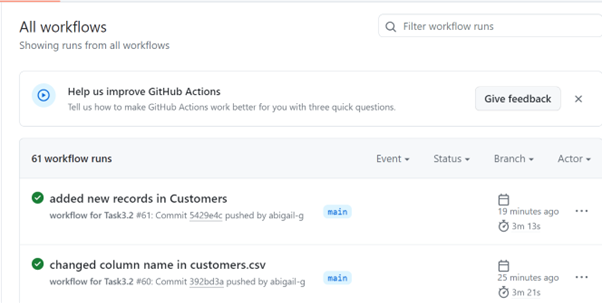


Figure 13: Workflow Timestamps as evidence of a successful workflow

# Data Analysis and Reporting with Quarto in R

### Distribution of product ratings from 0 to 5 coloured by category name

Information regarding product ratings, sales revenue, customer distribution, and order status was obtained through the establishment of a connection to an SQLite database and the execution of SQL queries. Subsequently, data visualisation techniques were used to identify the distribution and patterns within the data, offering insights into e-commerce operational dimensions.

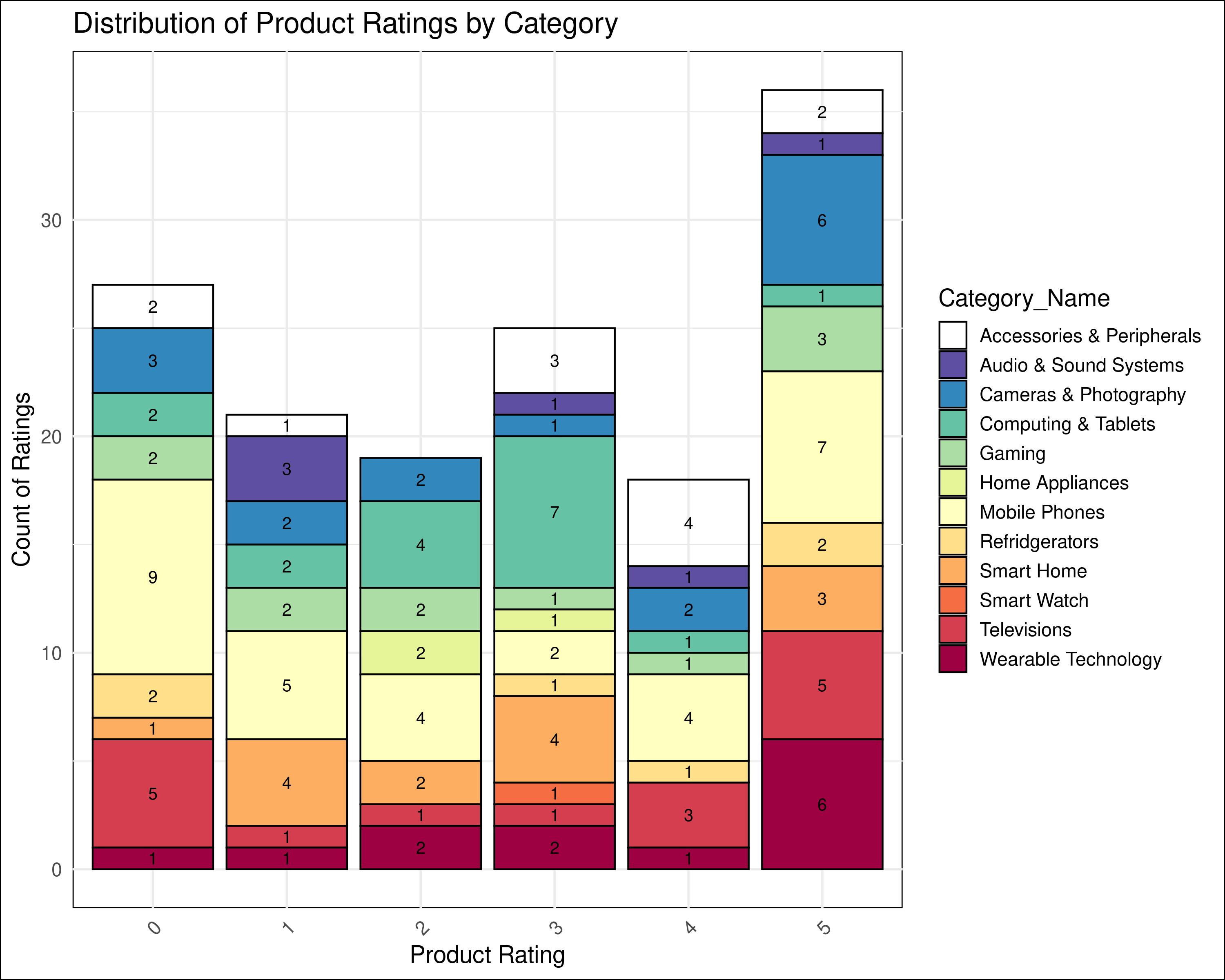


Figure 14: Distribution of Product Ratings by Category

### Top 5 Products Sorted by total Revenue

After joining the “Order\_Items”, “Discounts”, and “Products” tables, the derived value, which is the “Total Revenue”, is calculated by multiplying “Order\_Item” quantity and “Product\_Price” and deducting it with “Discount\_Amount.

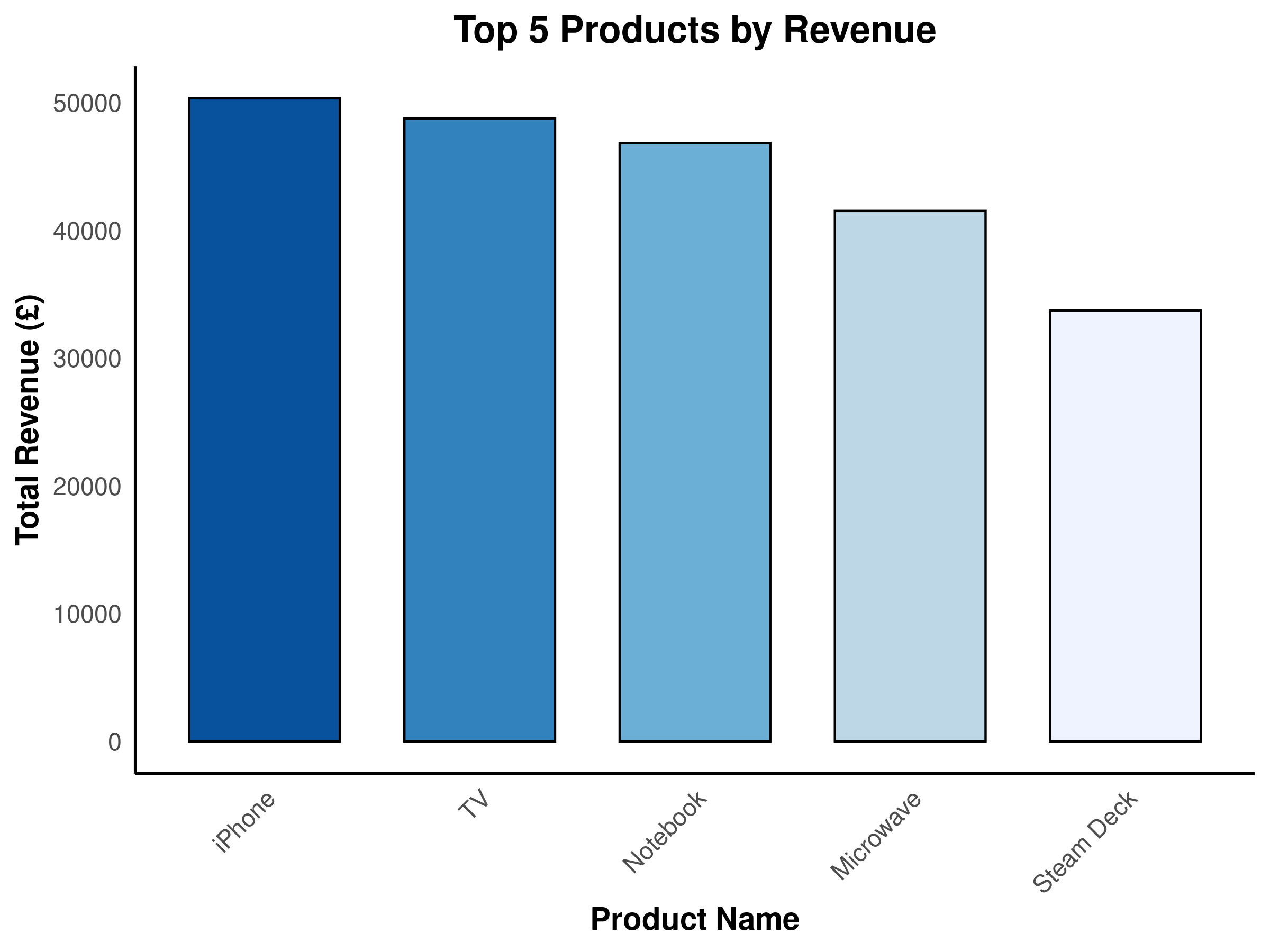


Figure 15: Top 5 Products by Revenue

### Distribution of customers by phone country codes

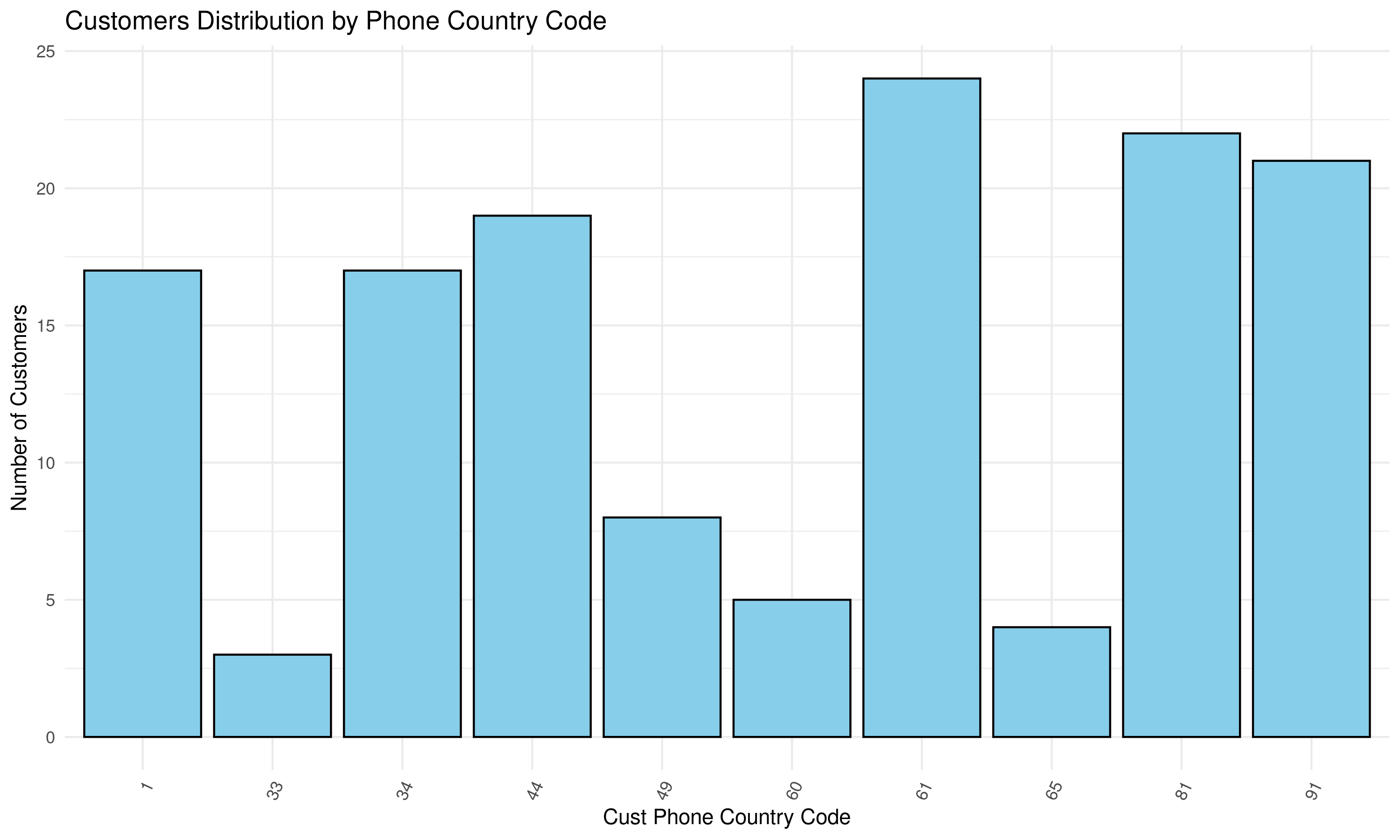


Figure 16: Distribution of Customers by Phone Country Code

### Order status count of shipped, processing, delivered and cancelled orders

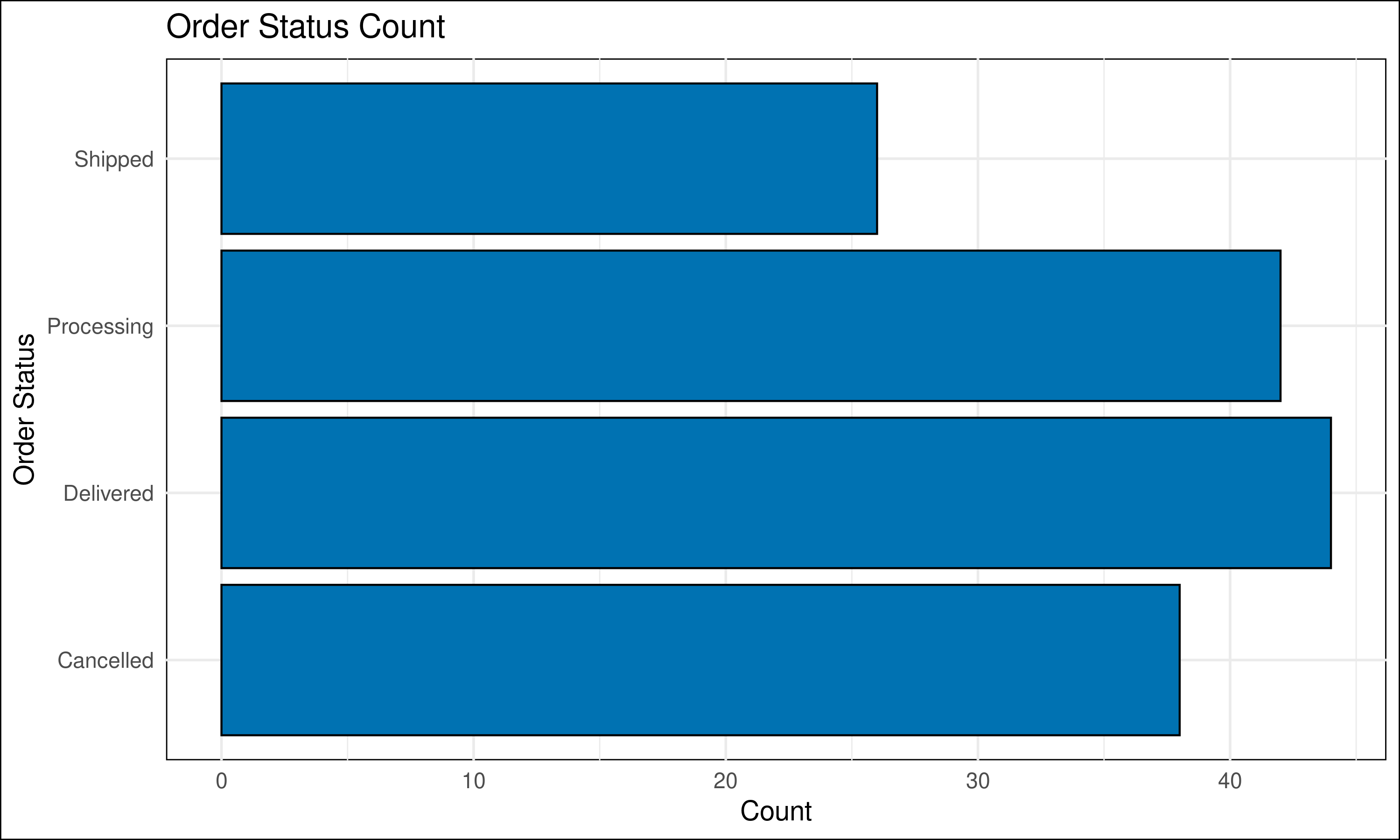


Figure 17: Order Status Count

# Conclusion

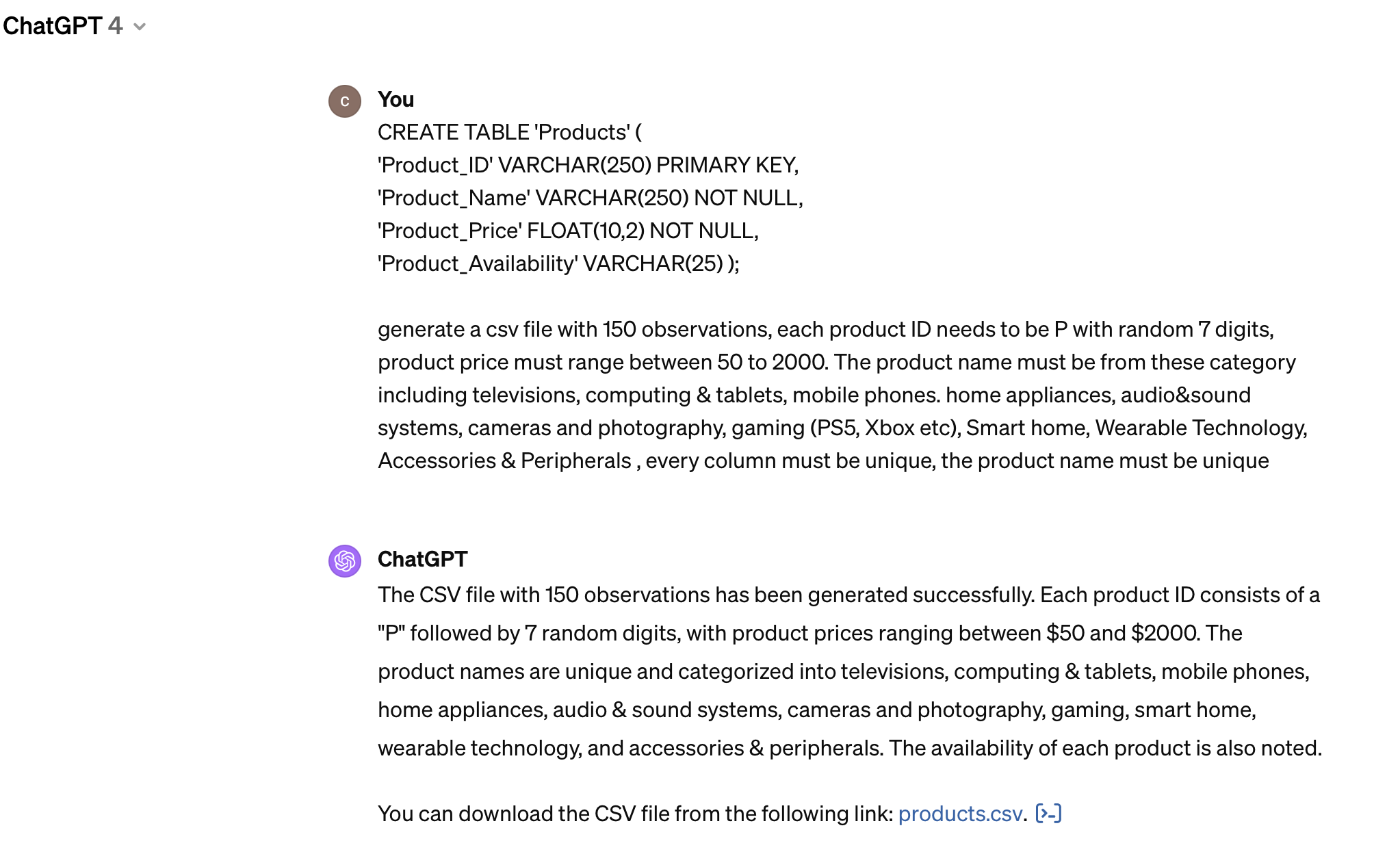
Upon completion of this project, valuable insights are gained regarding the challenges and methodologies associated with an e-commerce context, including the design of databases, analysis of data, and presentation of findings in a clear and impactful manner. Ultimately, this project serves as a reference for future e-commerce projects in particular and data-driven projects in general, providing valuable insight into the modern data management landscape.

# References

1. To be redirected to the project repository click [here](https://github.com/abigail-g/E-commerceDB)

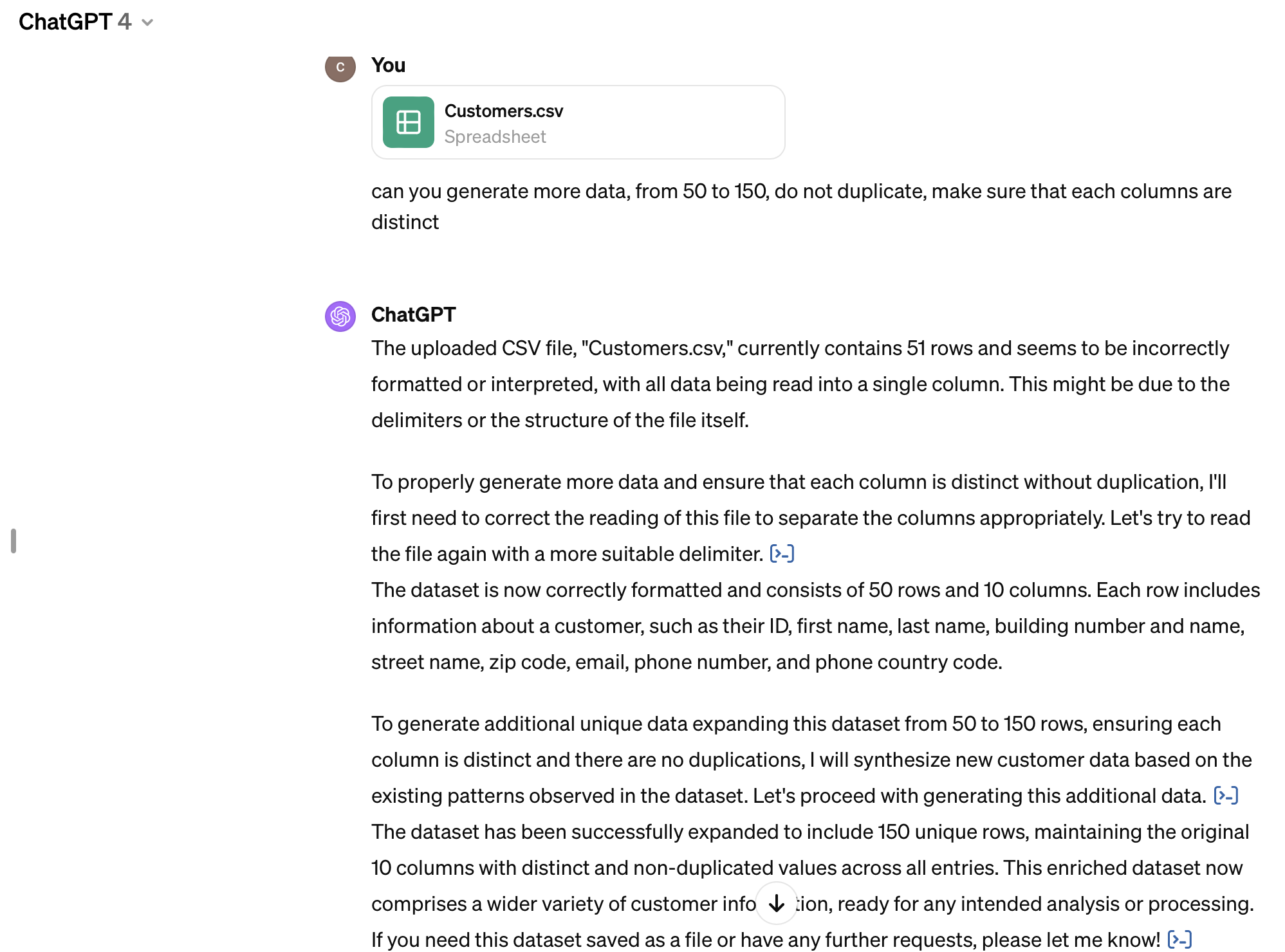
# Appendices

## Appendix 1



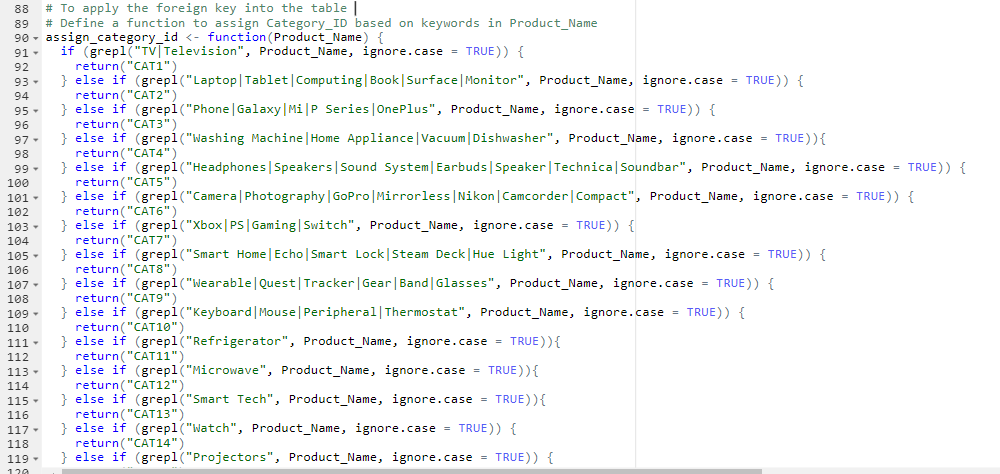
ChatGPT\_1

## Appendix 2



ChatGPT\_2

## Appendix 3



Data Update