WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

Master's Programmes: Group Assignment Cover Sheet

Student Numbers: Please list numbers of all group members	5583010,1863795, 5586298, 5558022, 5588532, 5570533
Module Code:	IB9HP0
Module Title:	Data Management
Submission Deadline:	12:00 (UK time) Wednesday 20th March 2024
Date Submitted:	Tuesday 19th March 2024
Word Count:	1973
Number of Pages:	15
Question Attempted: *(Question number/title, or description of assignment)*	Group Assignment
Have you used Artificial Intelligence (AI) in any part of this assignment?	Yes (Only using for generating synthetic data in task 2.1)

Academic Integrity Declaration
We're part of an academic community at Warwick. Whether studying, teaching, or researching, we're all taking part in an expert conversation which must meet standards of academic integrity. When we all meet these standards, we can take pride in our own academic achievements, as individuals and as an academic community.

Academic integrity means committing to honesty in academic work, giving credit where we've used others' ideas and being proud of our own achievements.

In submitting my work, I confirm that:

- I have read the guidance on academic integrity provided in the Student Handbook and understand the University regulations in relation to Academic Integrity. I am aware of the potential consequences of Academic Misconduct.
- I declare that the work is all my own, except where I have stated otherwise.
- No substantial part(s) of the work submitted here has also been submitted by me in other credit bearing assessments courses of study (other than in certain cases of a resubmission of a piece of work), and I acknowledge that if this has been done this may lead to an appropriate sanction.
- Where a generative Artificial Intelligence such as ChatGPT has been used I confirm I have abided by both the University guidance and specific requirements as set out in the Student Handbook and the Assessment brief. I have clearly acknowledged the use of any generative Artificial Intelligence in my submission, my reasoning for using it and which generative AI (or AIs) I have used. Except where indicated the work is otherwise entirely my own.
- I understand that should this piece of work raise concerns requiring investigation in relation to any of points above, it is possible that other work I have submitted for assessment will be checked, even if marks (provisional or confirmed) have been published.
- Where a proof-reader, paid or unpaid was used, I confirm that the proof-reader was made aware of and has complied with the University's proofreading policy.

Upon electronic submission of your assessment you will be required to agree to the statements above

Ecomm Report

Group 15

Part 1: Database Design and Implementation

Task 1.1: E-R Diagram Design

We are basing our database design and workflows on a real-world e-commerce data environment, inspired by Amazon. The e-commerce company only uses its own courier.

The ER diagram is in 1^{st} normalised form.

1. We assume that 1 product must belong to 1 category, and a category must at least have 1 or more product(s).
2. Many categories can be the child of a parent category.
3. Each product can only have 1 advertisement, and each advertisement is always for 1 product.
4. Each product has a unique supplier, but a supplier can supply many types of products.
5. Each customer only has 1 address.
6. 'Order' is a relationship between 'Customer', 'Product' and 'Transaction' entities.
7. 1 order can have many products, and a type of product can be in multiple orders.
8. A customer can make multiple orders, but each order must belong to 1 customer.
9. Each order may or may not have a transaction. An order must only be paid in a single transaction, by credit or debit card.
10. The customer who makes the order is the one who pays as well.
11. Each order may or may not have a delivery. One delivery is made for only one order.

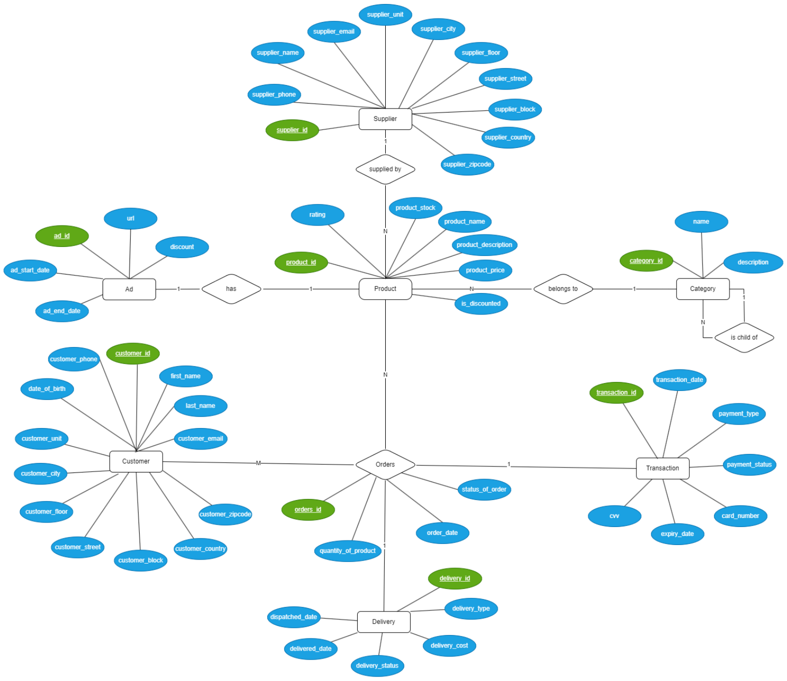


Figure 1: Final ER Diagram

Relationships between key entities are shown in Figure 2. We can tell that:

1. One-to-Many relationship: CUSTOMER and TRANSACTION, SUPPLIER and PRODUCT, CATEGORY and PRODUCT.
2. Many-to-Many relationship: CUSTOMER and PRODUCT
3. One-to-One relationship: TRANSACTION and DELIVERY, ADVERTISEMENT and Product
4. Recursive relationship: CATEGORY

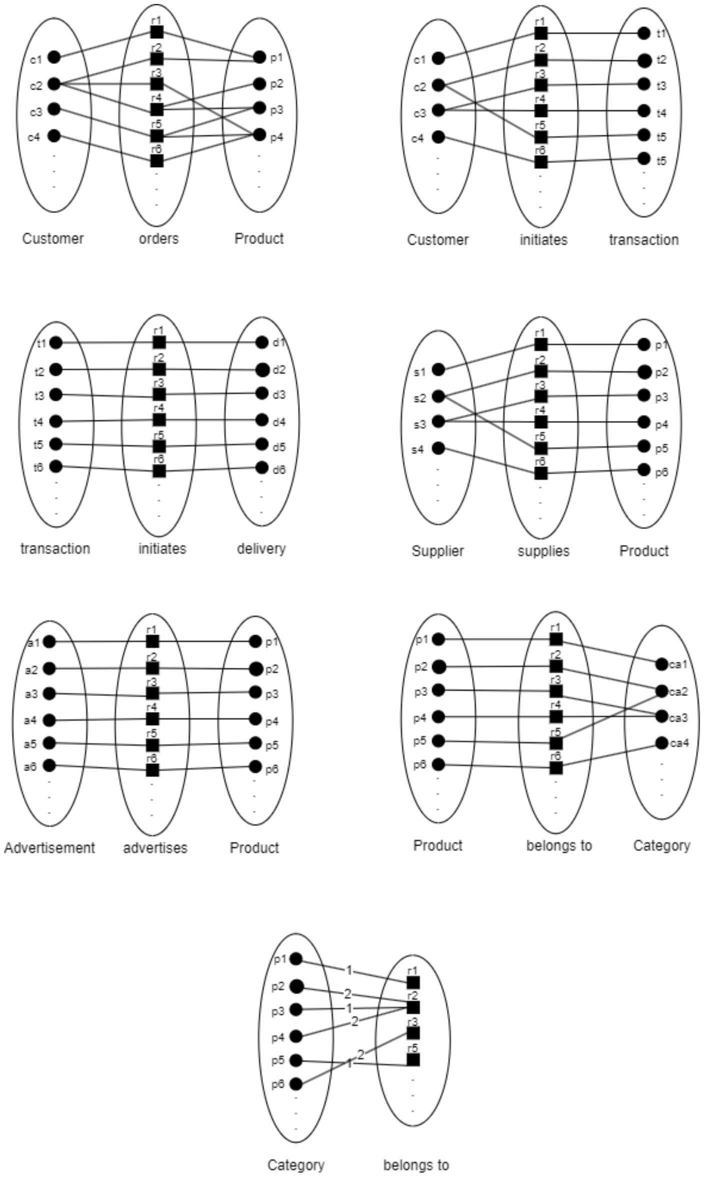


Figure 2: Relationship Set

Task 1.2: SQL Database Schema Creation

To convert 1st normalised form to 3rd normalised form, we have removed all partial and transitive dependencies in the entity tables.

As a result, we have obtained these tables in logical schema:

1. CUSTOMER (customer_id, first_name, last_name, date_of_birth, customer_phone, customer_email, customer_unit, customer_floor, customer_block, customer_street, customer_city, customer_country, customer_zipcode)
2. PRODUCT (product_id, supplier_id, category_id, product_name, product_description, product_price, product_stock, rating)
3. SUPPLIER (supplier_id, supplier_name, supplier_phone, supplier_email, supplier_unit, supplier_floor, supplier_block, supplier_street, supplier_city, supplier_country, supplier_zipcode)
4. ADVERTISEMENT (ad_id, product_id, ad_url, ad_start_date, ad_end_date, discount)
5. TRANSACTIONS (transactions_id, order_id, transaction_date, payment_type, card_number, expiry_date, cvv, payment_status)
6. CATEGORY (category_id, name, description, parent_category_id)
7. ORDER (order_id, customer_id, order_date, status_of_order)
8. ORDER_DETAILS (order_id, product_id, quantity)
9. DELIVERY(delivery_id, order_id, delivery_type, dispatched_date, delivered_date, delivery_status, delivery_cost)

We have made these assumptions:

1. Customer address is the same as their billing and delivery address, so it is only stored once in the Customer table.
2. Delivery cost only depends on delivery type (e.g. standard, premium), not the delivery address. We assume that cost of delivery type is fixed for whole nation-wide delivery.
3. Delivered_date is an input of the actual timedate when the parcel reaches the customer. It is not a calculated field (ETA) based on delivery type and order date, because there may be delays for which the parcel reaches customer. Delivered_date must be after dispatched_date
4. Delivery status is dependent on dispatched date and delivered date: “Pending” (both dispatched_date and delivered_date is NULL), “shipped” means the parcel is on the way (delivered_date is NULL), “delivered” means parcel has reached the customer.

In the database schema depicted in Figure 2, based on previous assumptions as well as the actual situation, the following data structuring conventions have been established:

1. Primary keys are designated in the Varchar format to accommodate future scalability and uniqueness, indicated by a yellow icon.
2. Date attributes are uniformly of the DATE type, adhering to the yyyy-mm-dd format.

3. Customer and supplier contact information, including phone numbers and emails, are unique across all entries.

4. Type attributes are constrained by the ENUM type, restricting values to a predefined set of keywords.

5. Descriptive text fields utilize the TEXT data type to avoid truncation.

6. The advertisement table presumes a singular, active advertisement per product with defined start and end dates, and an associated discount percentage. This discount is recorded as DECIMAL(6, 4) to ensure precision up to four decimal places.

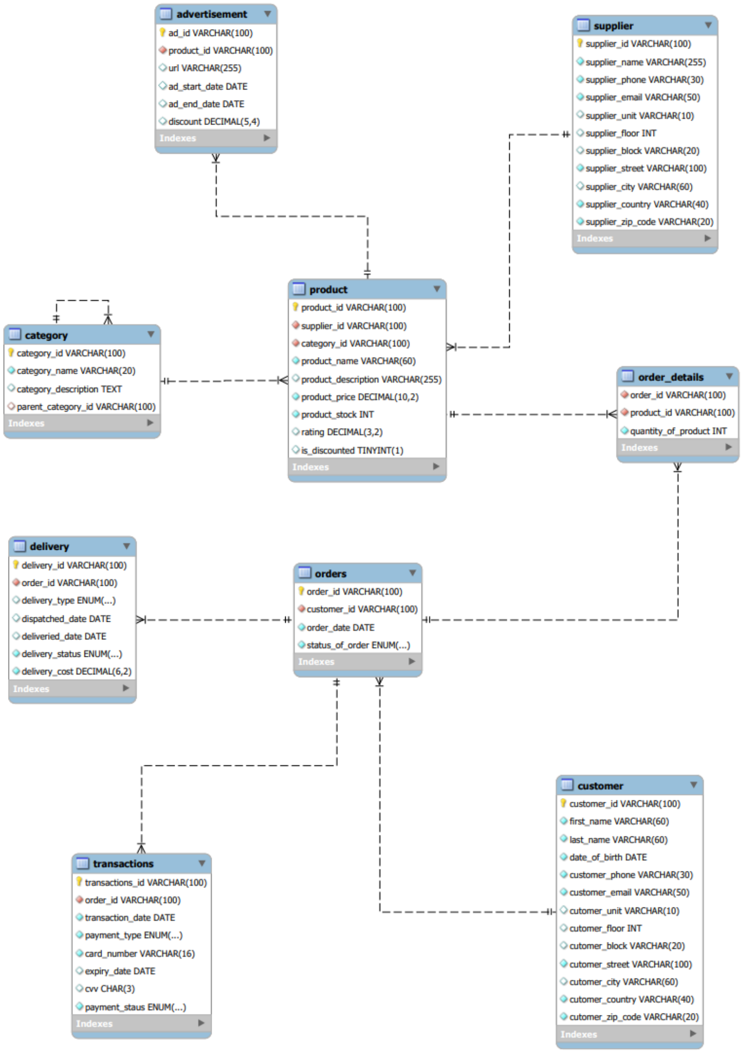


Figure 3: Table Schema

Part 2: Data Generation and Management

Task 2.1: Synthetic Data Generation

This section aims to generate synthetic data for an e-commerce business based on the structure of the database outlined in Part 1. To accomplish this task, it is crucial to thoroughly understand and identify the main components, including entities, attributes, primary keys, and foreign keys, within the data tables from the schema provided.

As per Part 1, there are nine tables for which data needs to be created in CSV format. These tables include advertisement, category, customer, delivery, order_details, orders, product, supplier, and transactions. AI and LLM tools such as ChatGPT and Mockaroo are utilized to generate the synthetic database. This is achieved by providing commands to these tools to obtain data. Additionally, we have decided to generate 1,000 samples for each table to ensure an adequate amount of data is available for analyzing the e-commerce business thoroughly.

Task 2.2: Data Import and Quality Assurance

After creating the data, we loaded it into our database and performed various procedures to ensure the quality and integrity of the data. It was essential to establish a strong foundation for the analysis phase. Through the integrated use of R and SQL codes, we preserved the consistency of data types and connections, verified the uniqueness of attributes, and validated foreign keys. This careful approach guarantees that our synthetic data faithfully mirrors a real e-commerce database.

Data Import Workflow

We initiated a structured workflow within RStudio and Posit Cloud for an organized management of our GitHub repository. With a complete set of tables ready for import, we aimed to create a robust e-commerce environment.

Ensuring Data Quality and Integrity

Database and Table Creation: Using R scripts, we set up tables according to our schema, ensuring the structure adhered to our design principles.

1. Uniqueness Constraints: Uniqueness constraints were placed on critical attributes like customer emails and phone numbers to eliminate duplicates.
2. Referential Integrity Checks: We confirmed that all foreign keys corresponded to existing primary keys in their respective tables.
3. Data Import Execution: We connected to the database for importing data. Errors encountered were logged and resolved, ensuring accurate data import.

This method not only facilitated the efficient transfer of our synthetic data but also ensured its consistency and reliability, providing a firm foundation for subsequent analysis and reporting.

Part 3: Data Pipeline Generation

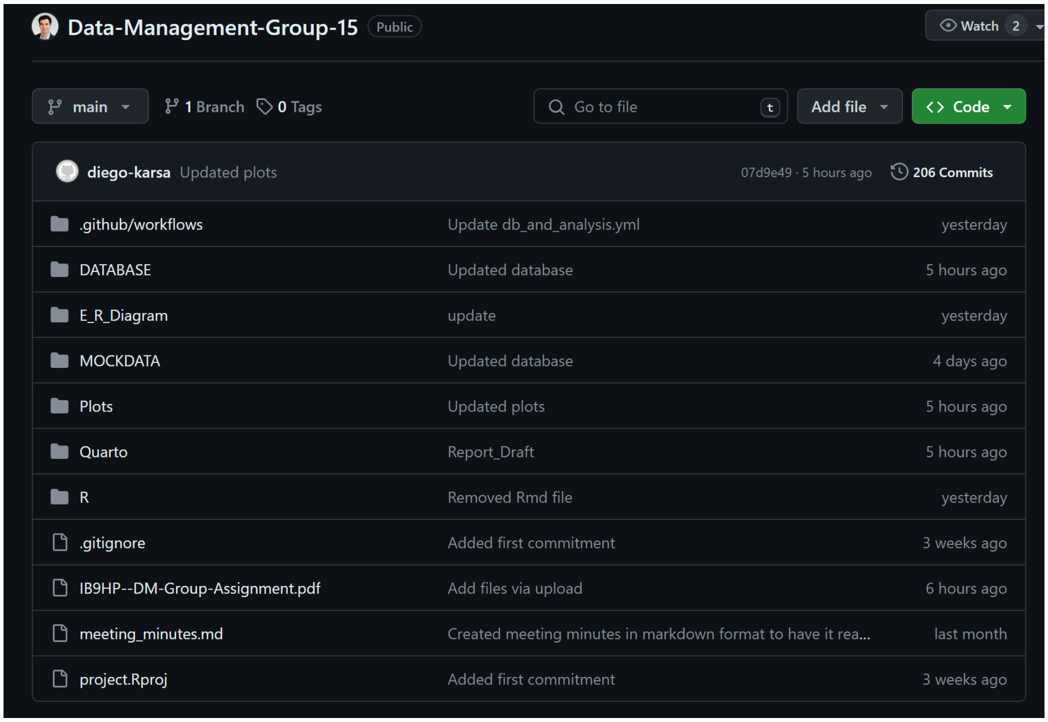
Task 3.1 GitHub Repository and Workflow Setup

Objective:

Utilize GitHub to streamline the database creation process and maintain up-to-date access for all team members.

GitHub Repository Structure:

We created ' github.com/diego-karsa/Data-Management-Group-15 ' repository for project management and version control. This repository contains all vital elements, including the SQL schema, setup scripts, query scripts, project documentation, and data analysis plots.



diego-karsa Updated plots 07d9e49 · 5 hours ago 206 Commits

.github/workflows	Update db_and_analysis.yml	yesterday
DATABASE	Updated database	5 hours ago
E_R_Diagram	update	yesterday
MOCKDATA	Updated database	4 days ago
Plots	Updated plots	5 hours ago
Quarto	Report_Draft	5 hours ago
R	Removed Rmd file	yesterday
.gitignore	Added first commitment	3 weeks ago
IB9HP--DM-Group-Assignment.pdf	Add files via upload	6 hours ago
meeting_minutes.md	Created meeting minutes in markdown format to have it rea...	last month
project.Rproj	Added first commitment	3 weeks ago

Figure 4: GitHub Repository

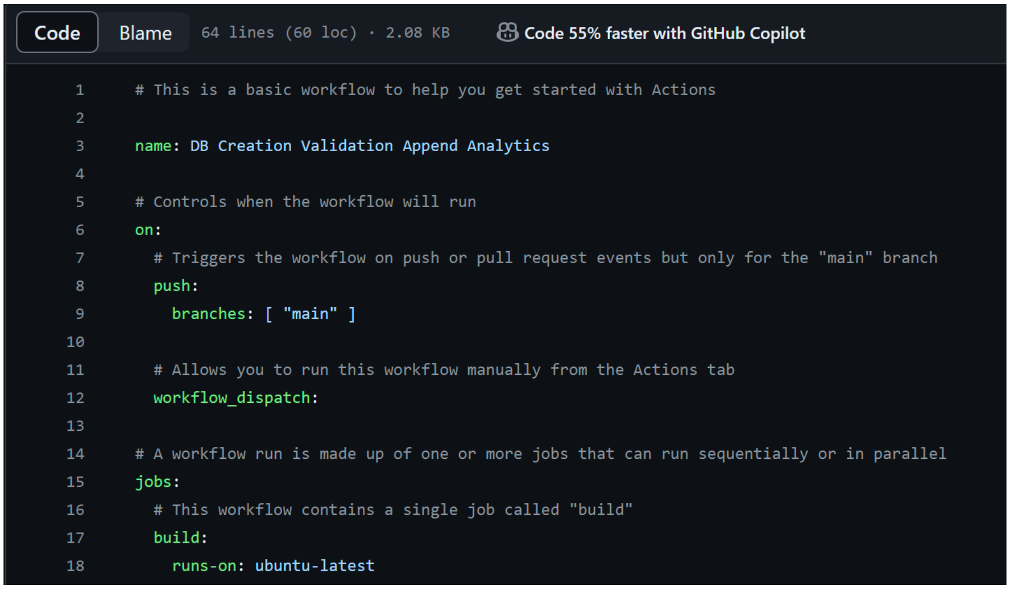
Task 3.2 GitHub Actions for Continuous Integration

Objective:

Implement GitHub Actions to automate crucial tasks like data validation, database updates, and initial data analysis.

Workflow Setup:

A Github action workflow named ‘db_and_analysis.yml‘ was implemented to automate our data pipeline. Triggered by push events or manually, this workflow facilitates data validation, updates, and analysis on an Ubuntu server.

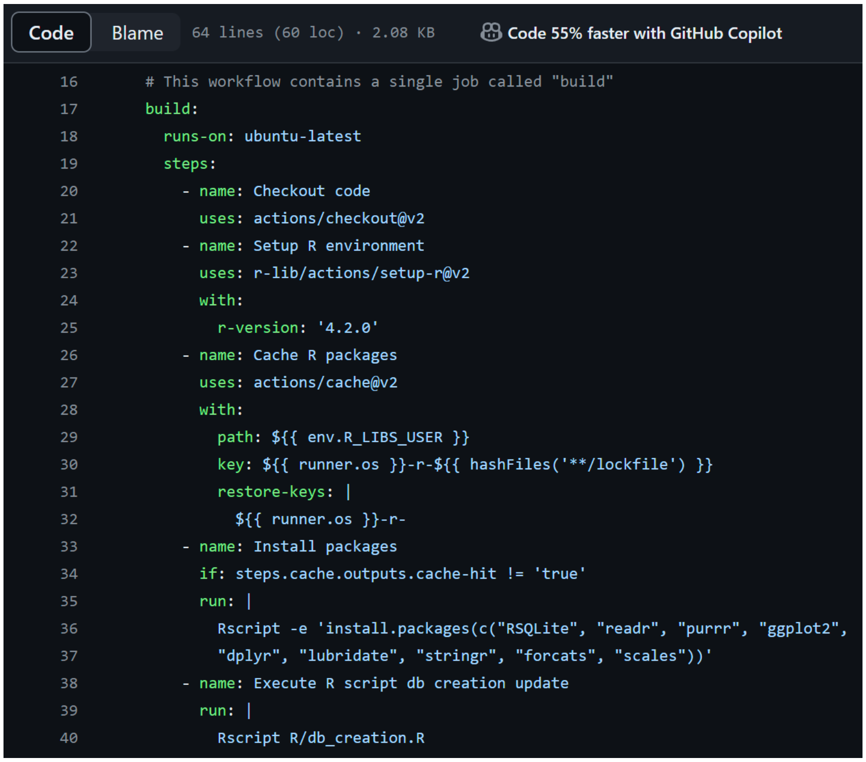

Code
Blame
64 lines (60 loc) · 2.08 KB
Code 55% faster with GitHub Copilot

```yaml
# This is a basic workflow to help you get started with Actions

name: DB Creation Validation Append Analytics

# Controls when the workflow will run
on:
  # Triggers the workflow on push or pull request events but only for the "main" branch
  push:
    branches: [ "main" ]

  # Allows you to run this workflow manually from the Actions tab
  workflow_dispatch:

# A workflow run is made up of one or more jobs that can run sequentially or in parallel
jobs:
  # This workflow contains a single job called "build"
  build:
    runs-on: ubuntu-latest
```

Figure 5: Workflow Setup

Workflow Tasks:

1. Setting up R environment: The latest repository code is retrieved, and the R environment is set up with necessary package caching.
2. Package installations and database creation: Required R packages are installed, and the ‘db_creation.R‘ script builds the database, followed by the addition of changes to the ‘DATABASE‘ folder.


Code
Blame
64 lines (60 loc) · 2.08 KB
Code 55% faster with GitHub Copilot

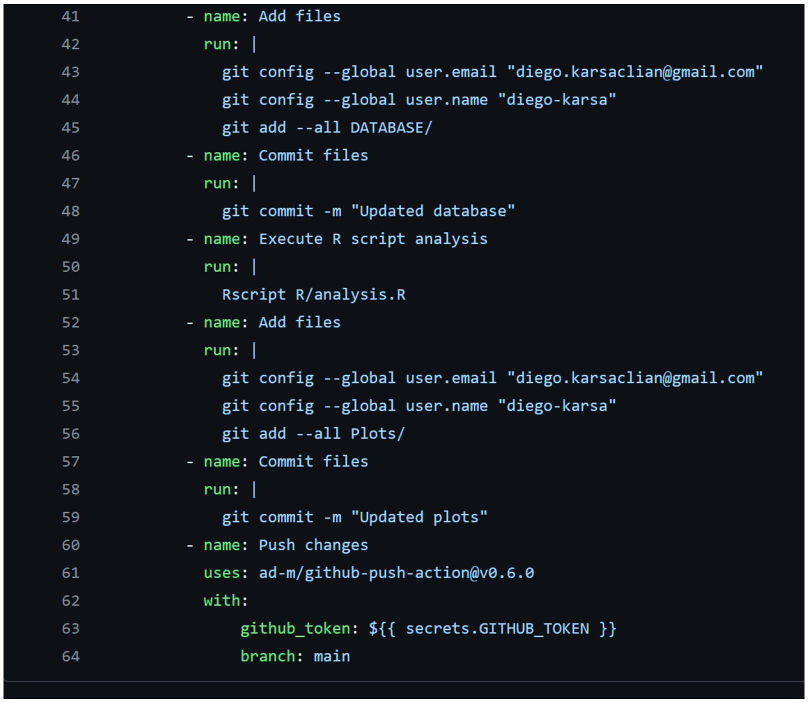
```
# This workflow contains a single job called "build"
build:
  runs-on: ubuntu-latest
  steps:
    - name: Checkout code
      uses: actions/checkout@v2
    - name: Setup R environment
      uses: r-lib/actions/setup-r@v2
      with:
        r-version: '4.2.0'
    - name: Cache R packages
      uses: actions/cache@v2
      with:
        path: ${{ env.R_LIBS_USER }}
        key: ${{ runner.os }}-r-${{ hashFiles('**/lockfile') }}
        restore-keys: |
          ${{ runner.os }}-r-
    - name: Install packages
      if: steps.cache.outputs.cache-hit != 'true'
      run: |
        Rscript -e 'install.packages(c("RSQLite", "readr", "purrr", "ggplot2",
        "dplyr", "lubridate", "stringr", "forcats", "scales"))'
    - name: Execute R script db creation update
      run: |
        Rscript R/db_creation.R
```

Figure 6: Workflow for Database Creation using RScript

3. Data analysis and visualizations: The ‘analysis.R‘ script runs for data analysis and visualization creation, with outputs added to the ‘Plots‘ folder and then to the repository.

4. Committing changes: Changes are committed with a descriptive message and pushed to the main branch.

Through this workflow, we have automated the data management process, enhancing the efficiency and reliability of our system, showcasing GitHub Actions as a powerful tool for continuous integration in data-driven projects.



```
      - name: Add files
        run: |
          git config --global user.email "diego.karsaclian@gmail.com"
          git config --global user.name "diego-karsa"
          git add --all DATABASE/
      - name: Commit files
        run: |
          git commit -m "Updated database"
      - name: Execute R script analysis
        run: |
          Rscript R/analysis.R
      - name: Add files
        run: |
          git config --global user.email "diego.karsaclian@gmail.com"
          git config --global user.name "diego-karsa"
          git add --all Plots/
      - name: Commit files
        run: |
          git commit -m "Updated plots"
      - name: Push changes
        uses: ad-m/github-push-action@v0.6.0
        with:
            github_token: ${{ secrets.GITHUB_TOKEN }}
            branch: main
```

Figure 7: Workflow for Data Analysis using R and Pushing Changes

Part 4: Data Analysis and Reporting with Quarto in R

Part 4.1: Advanced Data Analysis in R

Objective:

Our objective was to conduct comprehensive data analysis on the e-commerce data, focusing on time-dependent quantitative insights. This involved using advanced R packages for data manipulation and visualization, such as dplyr, ggplot2, and lubridate.

Analysis Overview:

The analysis encompassed several key areas:

- Sales Insights: We analyzed sales by location and category in the latest year, identifying top-performing cities and categories.
- Order Fulfillment and Value: We evaluated the average time until order fulfillment by quarter and the average order value, providing insights into operational efficiency.
- Product Performance: Analysis of top-rated products by category offered a view into customer preferences.

Visualizations:

Using the ggplot2 library, we generated visualisations for each analysis domain, such as bar plots and line graphs. The plots illustrated periodic trends, including sales patterns and client expansion, and were specifically built to be effortlessly updated when fresh data is acquired through our data pipeline.

These insights provide vital information for making strategic decisions, aiding in the optimisation of sales strategies, product offerings, and customer engagement initiatives.

Part 4.2: Comprehensive Reporting with Quarto

Objective:

The intent of this section is to detail the insights derived from a sophisticated analysis of our e-commerce data over the years 2020-2023, employing data visualizations from part 4.1, to communicate the findings effectively to stakeholders.

Key Findings:

Average Time Until Order Fulfillment:

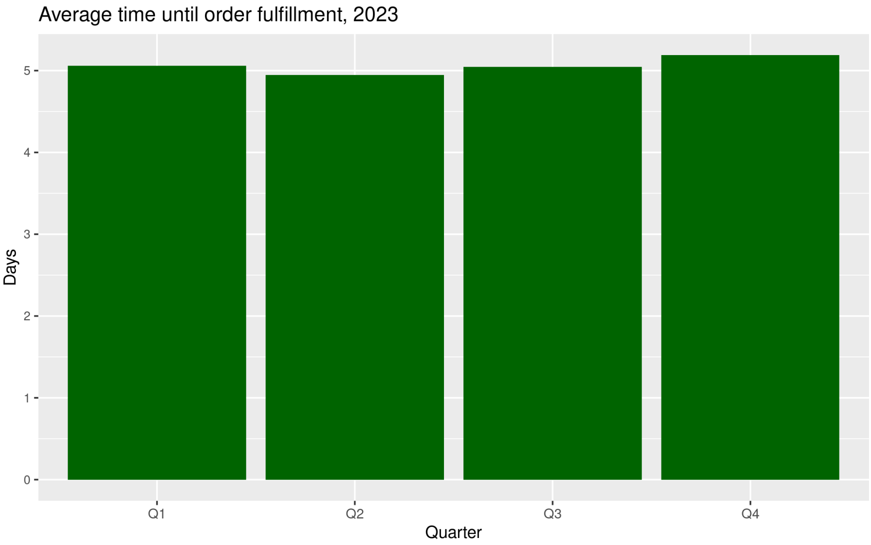


Figure 8: Average Time Until Order Fulfillment

The ‘Average time until order fulfillment, 2023’ graph demonstrates a consistent delivery performance across all four quarters, indicating an effective and stable logistics system. This consistency is crucial for customer satisfaction and retention.

Average Order Value:

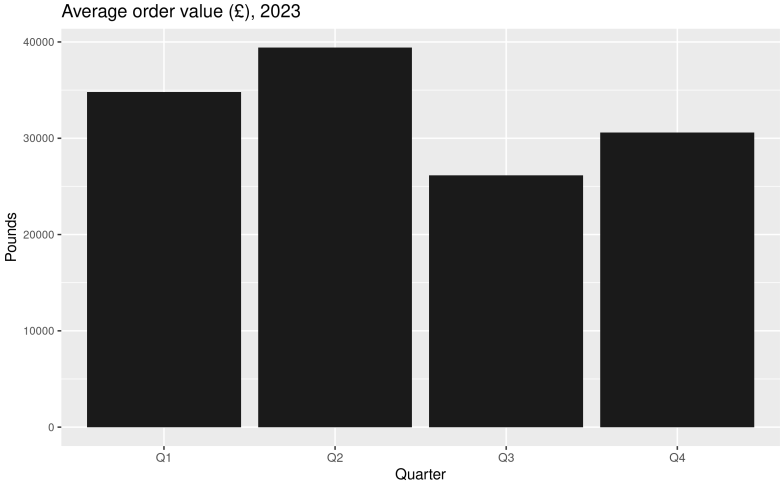


Figure 9: Average Order Value

Our analysis of the 'Average order value (£), 2023' highlighted seasonal fluctuations, with Q1 and Q4 experiencing higher values. This pattern suggests that promotions or seasonal demand are influencing shopping behavior, guiding marketing strategies for these periods.

Cumulative Customer Growth:

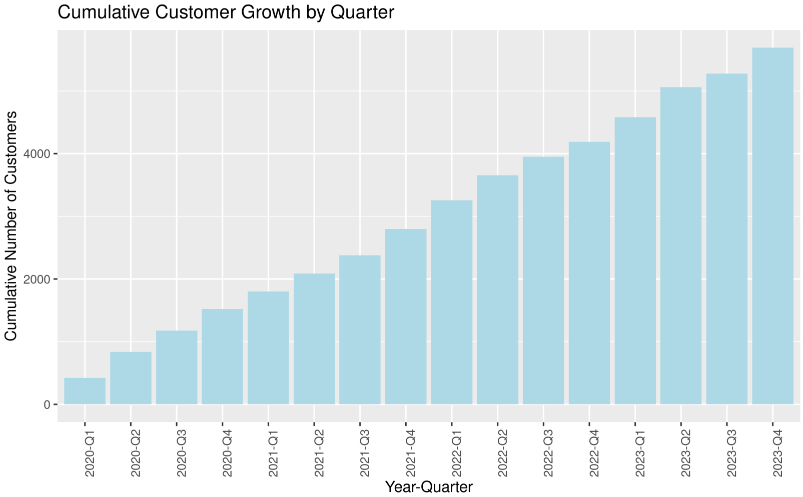


Figure 10: Cumulative Customer Growth

The 'Cumulative Customer Growth by Quarter' plot reveals a steady increase in our customer base. The consistent upward trajectory reflects successful marketing efforts and the growing appeal of our platform.

Age Distribution of Customers:

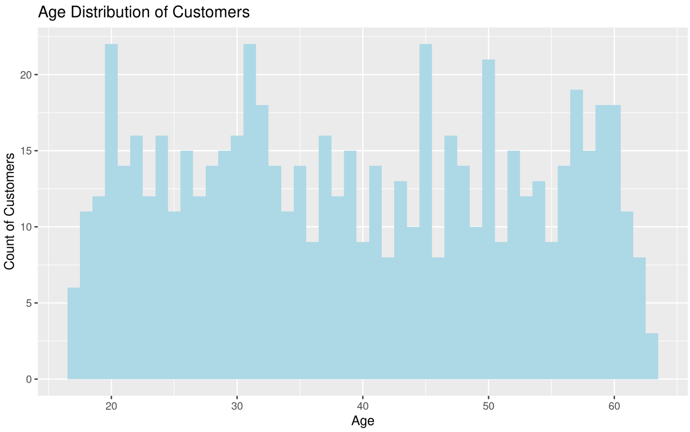


Figure 11: Age Distribution of Customers

The 'Age Distribution of Customers' histogram indicates a broad customer base, with noticeable concentrations around certain age groups. This diversity suggests that our products have a wide appeal, and also highlights key demographic segments for targeted marketing.

Monthly Sales Over Time:

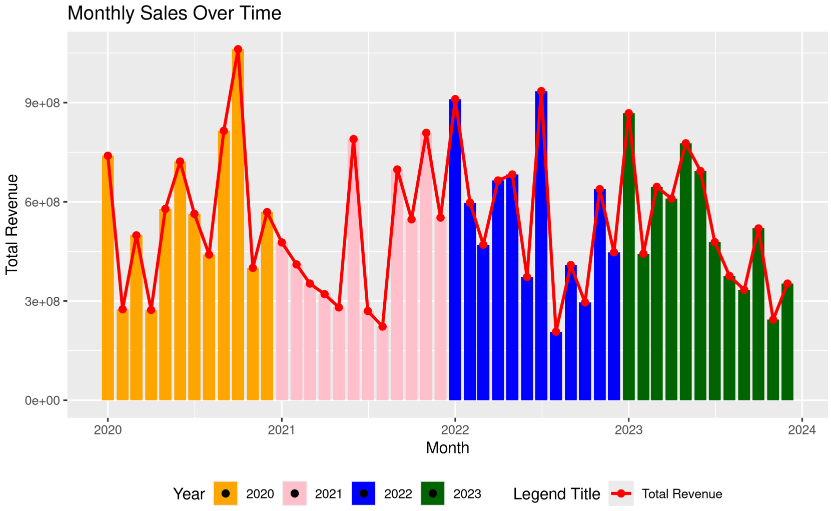


Figure 12: Monthly Sales Over Time

The 'Monthly Sales Over Time' line graph provides a dynamic view of sales performance, with peaks typically in Q4 each year, underscoring the impact of holiday seasons and possibly annual sales events.

Sales by Category:

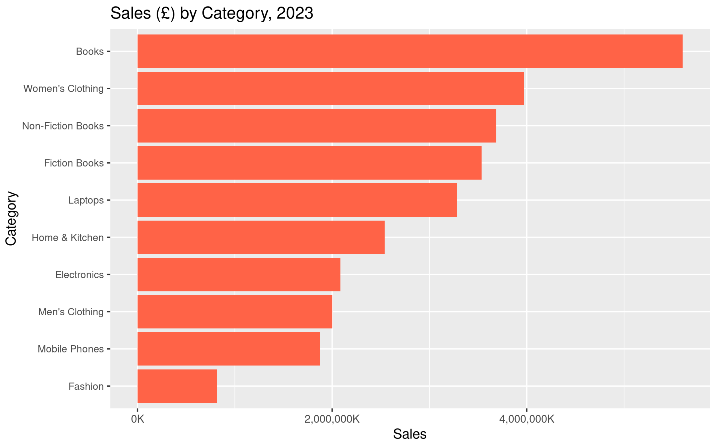


Figure 13: Sales by Category

Significant insights come from the 'Sales (£) by Category, 2023' graph, showing that certain categories, like Books and Women's Clothing, are leading sales. This indicates product categories that could be prioritized for stock and promotional activities.

Top 10 Cities by Sales:

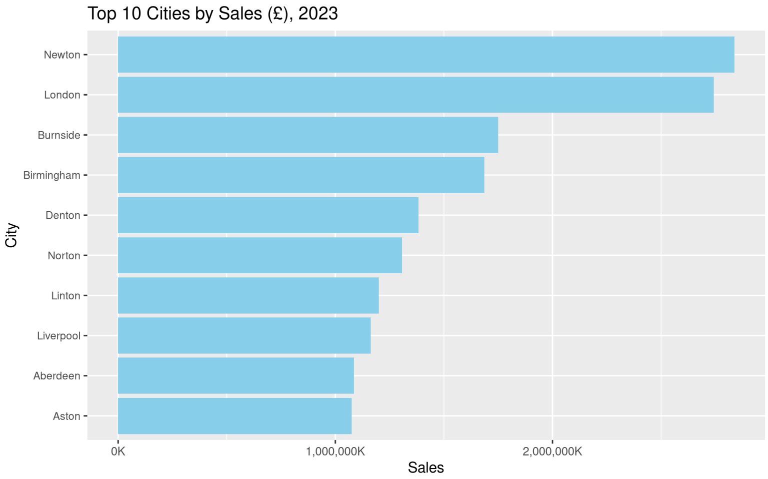


Figure 14: Top 10 Cities by Sales

In the ‘Top 10 Cities by Sales (£), 2023’, we observe that sales are concentrated in certain urban centers, suggesting areas where market penetration could be enhanced or where distribution networks might be optimized.

Average Rating by Category:

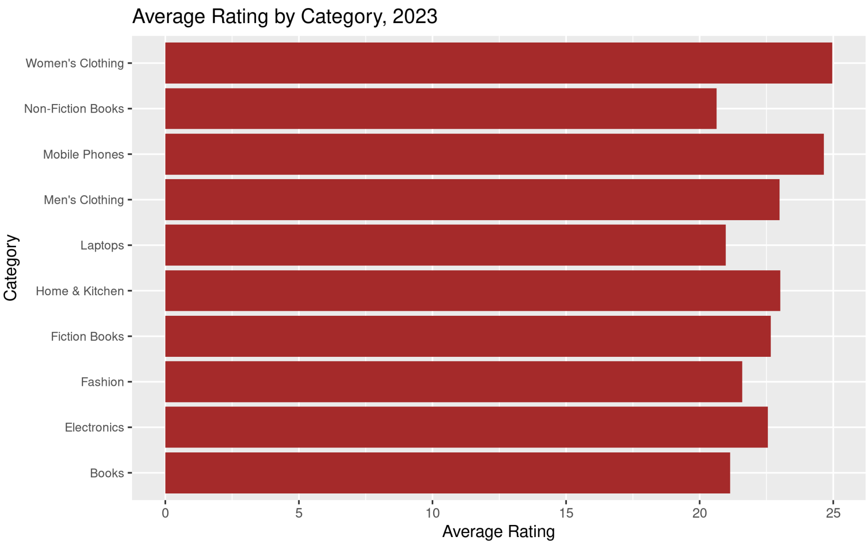


Figure 15: Average Rating by Category

Finally, the ‘Average Rating by Category, 2023’ plot illustrates customer satisfaction levels across different product categories, with some categories showing high ratings, which could inform inventory decisions and highlight areas for quality improvement.

Conclusion:

The analysed data presents a comprehensive view of the company’s operations throughout the years and focuses on key figures in the year 2023. The insights provide a solid foundation for strategic planning, focusing on customer engagement, sales optimization, and product management for the upcoming year.