# Chris Matthews Formative Piece 1

1.

I created three fictional datasets based on clothing sales from a shop, website, and app (figure 1). I then performed a data quality assessment.

Below are the issues identified for each data quality dimension:

Accuracy

* Quantities and prices are outside of normal range (figure 2). The data no longer reflects reality.
* Customer names had empty characters at the start (figure 3)
* Email addresses incorrectly formatted (figure 4). This can hinder analysis and reporting.
* Incorrect primary keys. Indicates unverified relationships and could break referential integrity.

Completeness

* NULL values (figure 5). Necessary information is missing from an order.
* Customer transaction method was not included. This will be added into the consolidated schema.

Consistency

* Inconsistent date formatting (figure 6). This can cause variations in how the data is stored making it harder to interpret results.

Uniqueness

* Duplicates. One table had been loaded twice resulting in double records (figure 7)

Timeliness

* Although I cannot measure the operational latency of the data inserted into the table verses expectations, the date ranges suggest this is a batch load that would not be timely enough for near real time (NRT) sales analysis.

The next step was to analyse the scale of the issues (figure 7) and fix them. For missing prices, I used an average value for the specific product. Given the small number of records affected by this issue this was an appropriate strategy. For the double loaded table, I assumed that this was an error and removed the duplicate values. I resolved formatting issues with trim functions. NULL values affecting primary keys were small volumes and removed all together.

Finally, I consolidated the datasets into one table and cleaned the data integrity issues (figure 8). I used a Jupyter notebook to run exploratory data analysis (EDA) on the original files to demonstrate a second method using the pandas, seaborn and matplotlib libraries in python (figures 9 and 10). This method is preferable when you want to present the issues to an audience and graphically display outliers or distribution using more advanced graphs such as box plots. I then re-ran this analysis on the cleansed file to confirm the data was clean (figures 11 and 12).

After the analysis I concluded that all the issues identified were in the web and app files, therefore transactions from these systems should be investigated by an ingestion team and resolved. I found the Retail data to be dependable. Pricing errors are isolated to the ‘Hats’ product group and could be due to configuration errors setting up the product in the dim product table.

2.

I designed a schema for a fact\_orders table in GCP BigQuery and uploaded the CSV data (DDL SQL Figure 13, Data Figure 14). BQ does not enforce primary keys, instead I added NOT NULL on order\_id with a description that conveys uniqueness. I then created dimension tables (Figures 15, 16) for dim\_product and dim\_date to show how star schema principals can be used to denormalise the data, improve query performance, reduce storage costs, and avoid redundancy by reducing the need for duplicated data or extra fields in one big computationally expensive table.

Figure 1 – Three source files

A screenshot of a computer

Description automatically generated

Figure 2 – Inaccurate prices and quantities

A screenshot of a spreadsheet

Description automatically generated

Figure 3 – Leading spaces

A screenshot of a computer

Description automatically generated

Figure 4 – Email formatting

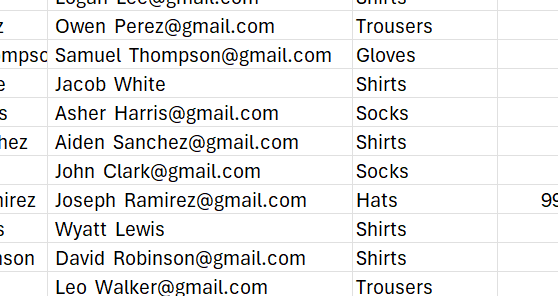


Figure 5 – NULL values

A white background with black numbers and letters

Description automatically generated

Figure 6 – Inconsistent date formats

A white background with numbers and letters

Description automatically generated

Figure 7 – Data quality metrics analysed on the consolidated file.

A screenshot of a spreadsheet

Description automatically generated

Figure 8 – Cleaned file.

A screenshot of a computer

Description automatically generated

Figure 9 – Null counts on uncleansed data using python.

A screenshot of a computer screen

Description automatically generated

Figure 10 – distribution of prices to find outliers.

A graph with numbers and a line

Description automatically generated

Figure 11 – corrected NULL checks.

A white rectangular object with black text

Description automatically generated

Figure 12 – Correct pricing after applying an average value for incorrect prices.

A graph of a distribution of unit prices

Description automatically generated

Figure 13. DDL to create fact\_orders table in GCP BigQuery.

A screen shot of a computer code

Description automatically generated

Figure 14. CSV loaded into the table.

A screenshot of a computer

Description automatically generated

Figure 15. DIM DATE DDL SQL.

A computer code on a white background

Description automatically generated

Figure 16. DIM PRODUCT DDL SQL.

A screen shot of a computer code

Description automatically generated