Data Integration Design Workgroup

IE Masters Big Data & Business Analytics - Business Intelligence and Data Warehousing

**Group E**

Maria Joyce, Oday Almajed, Ana Chavarri Ceballos, António Teixeira De Sousa Crespo Carvalho, Maximiliano Franco Martin, Mark Fleming-Williams

**Dataset - Campaign Data**

Following the creation of the data warehouse model, it is required to integrate data and load it into the model, making it available for final users to retrieve and analyse it. The approach to a proper data integration process must include different details such as the extraction of data from data sources, data mapping process, data quality tracking and metadata definition. These details cannot be properly split by chronological order since they are all linked. As a general overview of the chosen strategy for data integration, the first steps included the analysis of data sources and understanding of the required adjustments to assure data quality. At the same time, the data mapping approach consisted in guaranteeing a correct match between the data sources and the target data warehouse model, enabling final users to retrieve and analyze the data correctly. This connection between source and target is demonstrated in an attached excel spreadsheet.

The selected approach will be further described to allow for a better understanding of the decisions made when running the process of preparing extracted data for the final valid data warehouse model.

Looking back to the data warehouse model, a logical order needed to be followed regarding various data preparation tasks. The chosen order of the activities was to complete the dimension tables, fact tables and the bridge table. The one exception to this order is the visits fact table, which was completed last after the bridge table; this was done because the visits table contained a foreign key from the Bridge table, and is thus dependent upon it. The overall order ensures that the least dependent entries are created first, and the most dependent (e.g. foreign keys, which must first be created in a separate table as a primary key) last.

i) Dimension tables

The one Dimension Table that required **Data Mapping** was the Product table. The reason for this is that the data from the two columns did not match neatly, with the product family appearing once for every three times the product name appeared; in this way the linkage was clear to a human reader but not to a computer. In order to remedy this issue, a data mapping was undertaken. The Family column was deleted, and a new column was produced using javascript which linked each product to its family, making the connections complete.

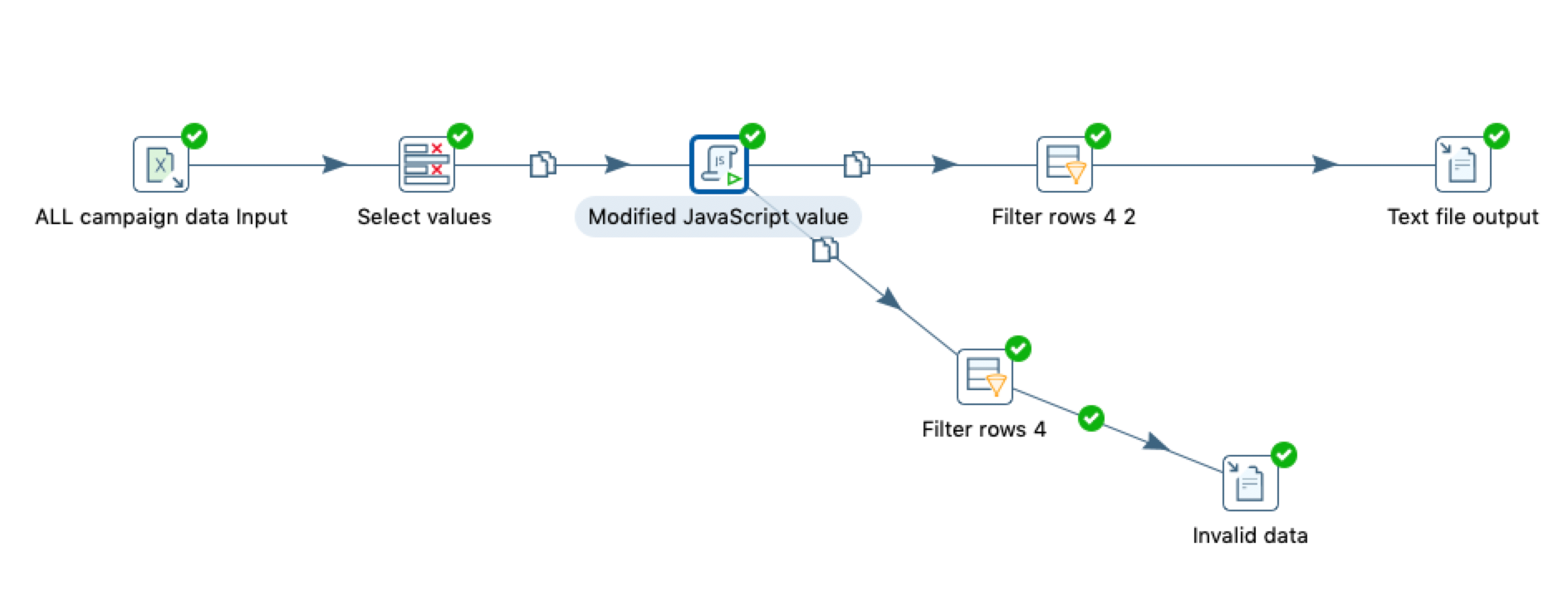
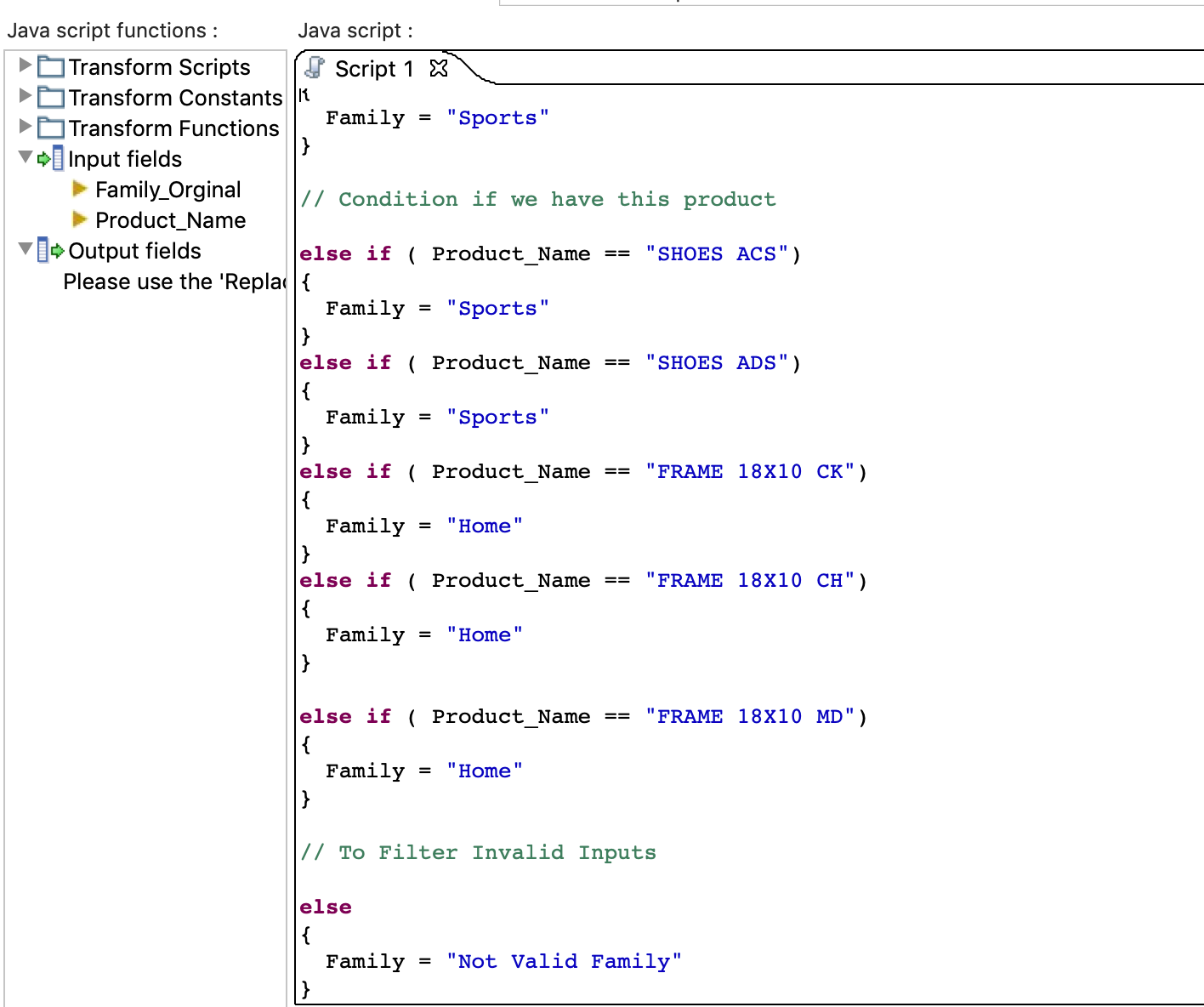


Fig 1. Javascript and cleansing of Product data

Once the Mapping had been completed the Data Quality tracking of the Dimension Tables could begin. The first step in this process was to **analyse the data**. This was undertaken across all the Dimension Tables. The procedure involved studying the source information, and the target destination of the information, and constructing a Microsoft Excel-based table which would act as a guide for when the loading process could finally occur.

Then the **data cleansing** could begin. This process saw the data being run through a series of parameters established in the previous assignment (see Appendix of Data Warehouse Modelling Workgroup assignment). This was undertaken in order to ensure that all the data was of high quality and did not contain errors such as spelling mistakes, unexpected minus numbers or null values. This process was undertaken using the Pentaho Data Integration program. The valid data was then retrieved and placed in position ready for loading. All valid records in the data set are written to a csv file. The invalid records are written to a separate file. The Naming Scheme for these files corresponds to the name of each worksheet in the dataset. e.g. invalid\_vists.csv and valid\_visits.csv. These bad records are not then subsequently loaded into the DB. For anyone who requires a list of bad or invalid records, they can simply inspect the contents of the invalid data files to see which records failed validation.

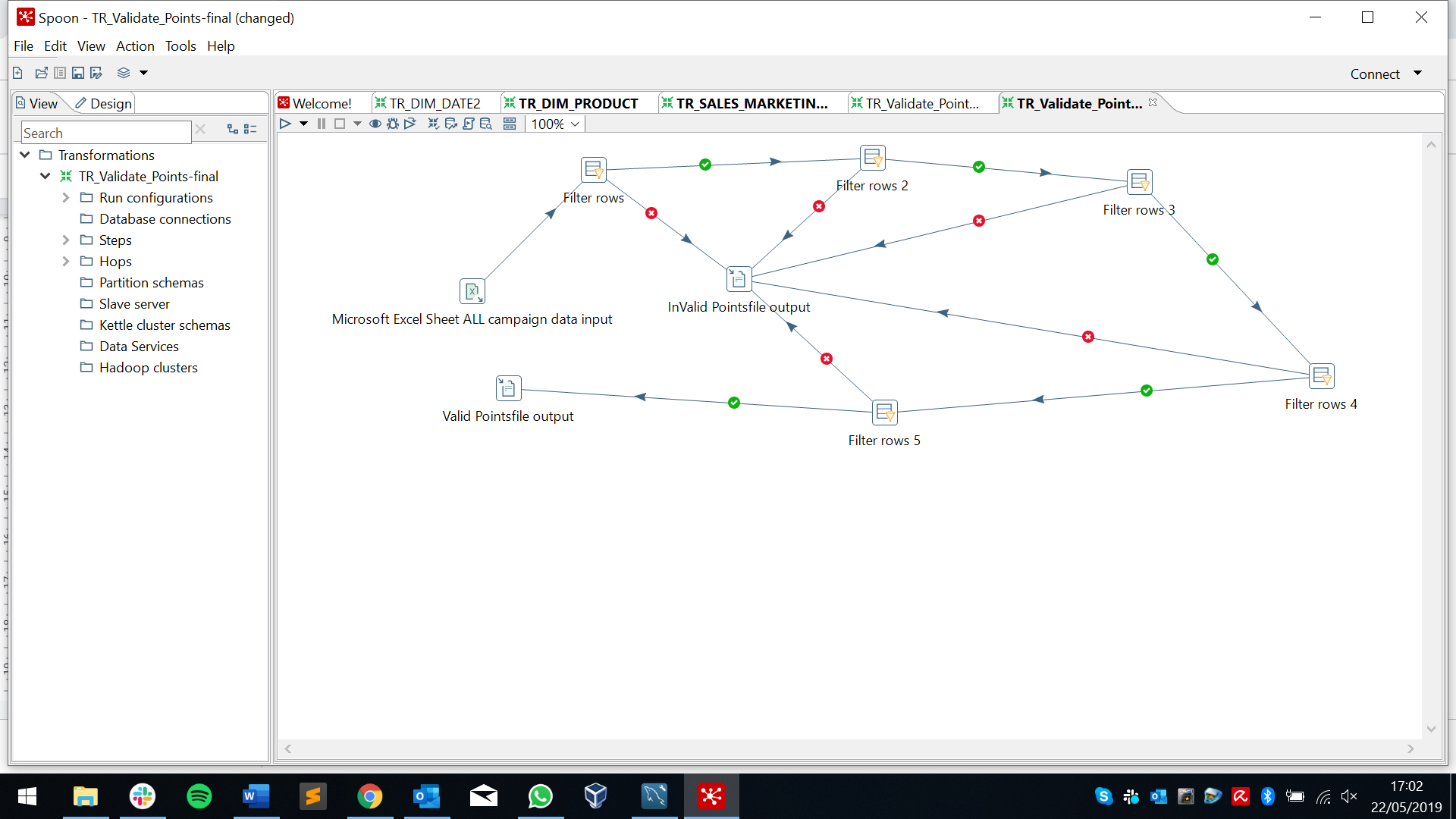


Fig 2. Example of data cleansing

ii) Fact Tables

*Visits*

The first fact table to be treated was the visits table. As the data source consisted of one unique spreadsheet, there were no steps regarding the integration of multiple sheets. Dates were checked regarding its format and validity to avoid wrong or missing dates. The number of visits per shopping center was confirmed to be positive or zero values. The number of visits per day matched perfectly the visits\_qty column, the main fact of the visits table.

*Sales and Marketing*

The sales and marketing fact table was one of the most critical tables of all, due to the existence of different data sources and the complexity of data preparation operations that would finally permit the loading of the data into the model. Starting with multiple sales sheets per shopping center and the same amount of sheets for impacts, it was required to carefully define all the transformations that would enable the creation of a unique table with all the information. To start, all the sales sheets were combined into one unique sheet with the exact same columns with all the products. In this sheet, shopping centers formed a unique column as a dimension of the sales. After this, the date validation occurred to assure it was in the correct format and without any null or missing values. To overcome having products as the header of this table, it was necessary to transpose them into a unique column, representing the number of sales per shopping center per day and per product. In the end, sales quantity was checked to avoid any negative or null values.

Following the same approach as for sales, impact´s sheets were combined into one single sheet having shopping centers as a unique column. Once again, dates were checked to avoid any kind of issues relating to integrity and quality. Products became a unique column to represent the number of impacts per product in a specific shopping center. Having a single column named “Impact Point” with both types of possible impacts stacked vertically, it was necessary to split it into two different columns in order to have one column for Information Point Screen and another one for Big Screen. As a result of the “row denormaliser” step, it was not possible to have the number of impacts for both types in the same column. This resulted in date multiplication, with each row having only one type of impact for one day for one shopping center and one product. Splitting the table into two different tables appeared to be a feasible solution to later eliminate all the null values. This was the chosen procedure, which then required to merge both files into a unique sheet with the number of impacts per day, product and shopping center for each impact type.

Combining both these transformations into a single output was the necessary step to get the final output as an individual table to be loaded into the data warehouse model.

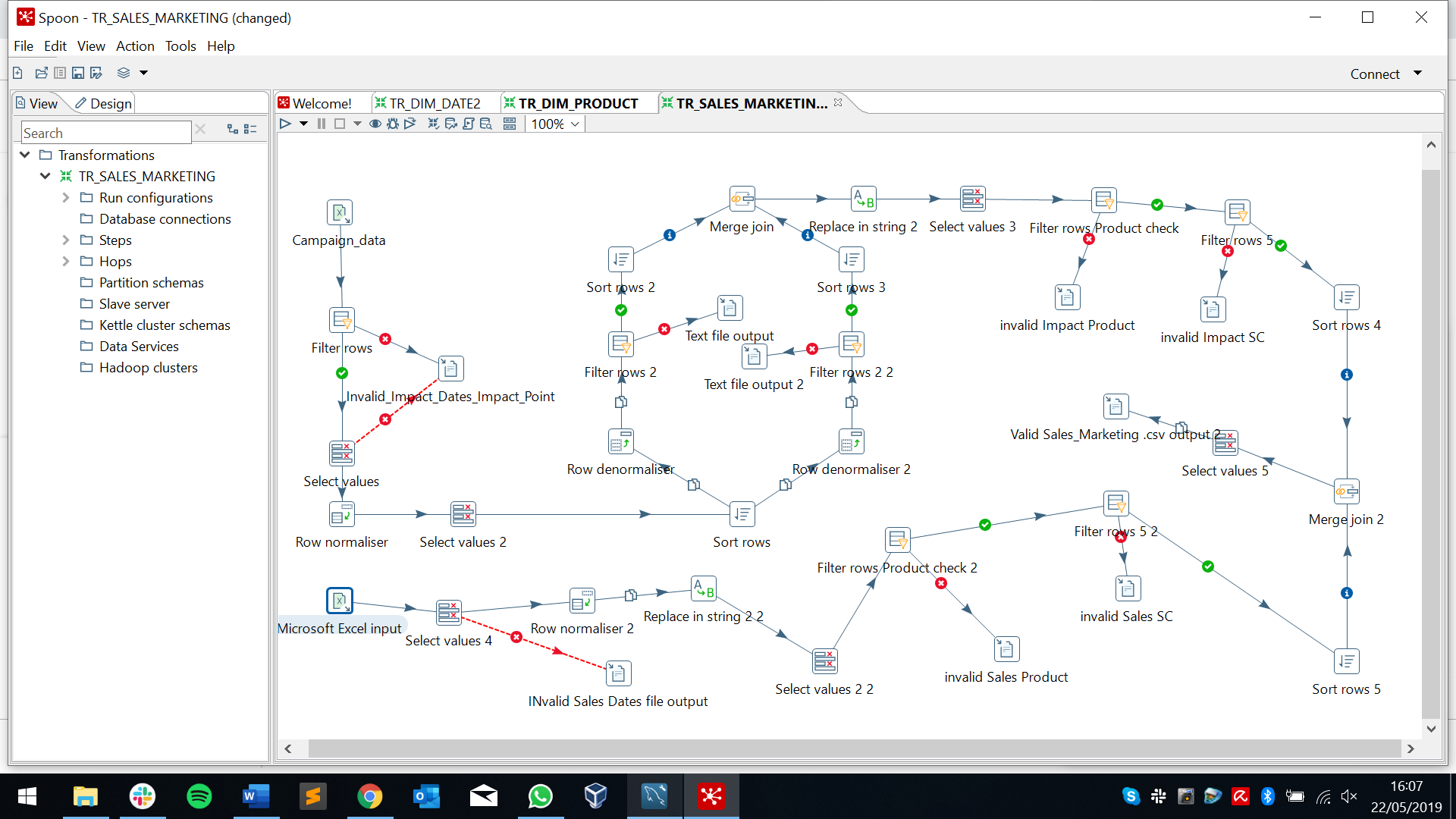


Fig 3. Process of sales & marketing data mapping

*Bridge table*

To populate the d\_bridge table there are three columns. The first is the primary key and the other two, d\_date and d\_shoppingcenter, are foreign keys from the Date and Shopping Center tables respectively.

Taking the date foreign key first. D\_date has two columns: id\_date and the date. There are 366 rows in this table and each date is unique. The bridge table problem is that the records in the bridge table must store the id of the date and not the date itself. To obtain the date id, the combination lookup/update pentaho step is used. For each date the corresponding ID date within the date table is used to populate the Bridge table.

Then looking at the shopping center dimension, it contains multiple rows for each shopping center name. Thus before using the combination lookup we must do two things: 1) identify the distinct shopping center names and save them to a CSV file 2) take the values in the CSV file and order them. Once this has been done, the combination lookup action can be undertaken.

Running the program

Running the Job Complete program sets several actions in motion.

First it runs the **Job DIM** file, which is the transformations of the Dimension Tables.

Within the Job DIM file is the Job data prep file. This runs the data quality validations of the various dimension tables – product, visits and shopping centers.

After the Job data prep file comes the transformations that load the dimensions into the database

Second it runs the **Job Fact Table Sales** file, which contains the transformations of the Fact Table for Sales and Marketing.

These are the same as the Dimension Tables, with data being first cleansed and then loaded.

Third it runs the **Job Bridge Table** file, which contains the transformation of the Bridge Table.

This is the same as the Dimension and Fact Tables.

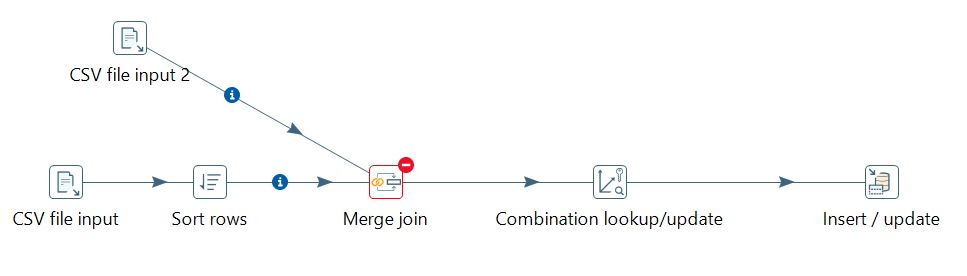
Fourth it runs the **Job Fact Visits** file, which contains the transformation of the Visits table.

Challenges:

Following the change of model between Assignment 2 and Assignment 3, the team decided to change its original plan around the Date table. In Assignment 2 the intention was to create ‘days of the week’ and ‘quarter’ columns in the Date table, but this plan was abandoned in Assignment 3 as it was decided that the new columns would be superfluous.

More generally Team E encountered some difficulties with this project. While the team feels it enjoyed substantial successes, unfortunately it was unable to complete the transformations, meaning that the Job will not work when it is initiated.

The major hindrances were within the Bridge dimension table and the Sales and Marketing fact table. Below is a screenshot of the merge join which ultimately failed within the Bridge table.



A solution we found but for which we ran out of time was to populate the date dimension and the shopping centre dimension from the same dataset.