SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

Data Warehouse Modelling Workgroup

Advertisement Data Set

Group B:

Cort Pons, Eduardo
Hung, Meng-Chen
Lee, Esther Chaelin
Mouawad, Ignacio
Raja, Varun
Satyal, Anup Raj
Villaveces, Juliana

Index

1 Note

Based on the recommendations made after assignment one, and after some deliberation as to an attribute on the fact table, a couple of changes were made to the initial dimensional model:

a) The label DIM_CHAR was changed to DIM_GENDER and id_CHAR was changed to id_GENDER.

b) The hit_efficiency attribute in the fact_effectiveness (fact table) was removed from the model. The hit_efficiency being a calculation of two fact_table attributes themselves, we believe that such a calculation is better made by a BI system (or user) as a part of a data analysis process rather than being kept as a part of the data warehouse.

The modified model with the changes mentioned above has been included in the zip file. The data load process for the fact table slowed down significantly once the hit_efficiency column was removed. The remaining processes were completed within five minutes, but we were not able to complete the data load into the fact table with these changes. In the case that the data load is equally slow for other users, we have included a backup file, in which the data loading was completed in three hours. The file includes the hit_efficiency column but with all its rows as zero.

2 Default Strategy

The extraction process was carried out using a single Transformation (TR_DATA_PREP), which took 69 seconds to finalize. We followed a 7-step process to extract and prepare information from the Platform tables:

i) Step one (Merge Platforms and Add Names): As each of the platform tables had identical columns (Date, Zip code, Product, Age, Gender, Likes, Prints, Hits), we merged all 4 platform tables to create a consolidated dataset. The names of each of the platforms corresponding to each of the tables were also added prior to the merge in order to ensure that Pentaho was able to look up the relevant prints and hits for each platform and/or post.

ii) Step two (Rename fields): We renamed the columns (e.g Product) to reflect the attribute names on our initial SQL model (e.g product_name) to ensure smooth running of the foreign key lookups during the fact table mapping process.

iii) Step three (Date Split): Since our date dimension in the initial model was structured in a way where day, month and year were segregated, the date column on the consolidated dataset was split using a calculation.

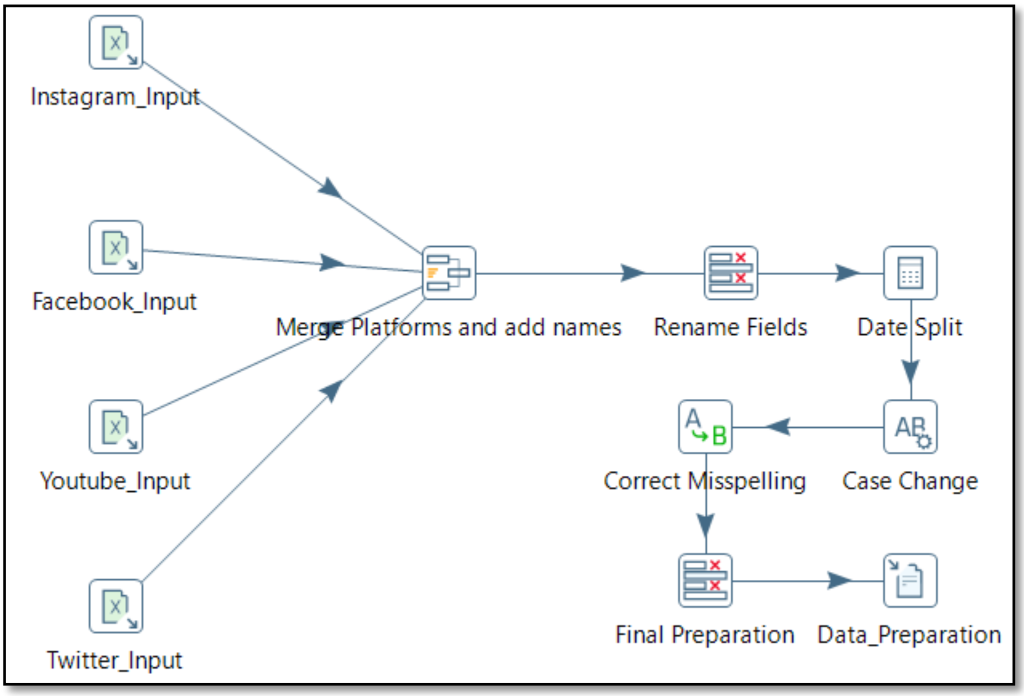
iv) Step four (Case Change): During step one, the platform names were added based on the name of the sheets on our initial (advertisement) dataset. As such, they were in uppercase format and contained whitespace after the name. This step was carried out to change the names to lowercases and remove any white spaces from the names. Since some of the other categories (e.g. product and gender) also had a mix between upper and lower cases, this step also converted all the mixed cases into lower cases to provide consistency across the database.

v) Step five (Correct Misspellings): This step improved the data quality by replacing the misspelled product name “sheatshirt” to “sweatshirt”.

vi) Step six (Final Preparation): In this step, metadata was added to the zip_code, and target_age_range attributes by configuring their types as Integer, Number, and String respectively.

vii) Step seven (Data Extraction): The final dataset was then extracted as a CSV file and exported into a Data Output folder for further use in the following processes.

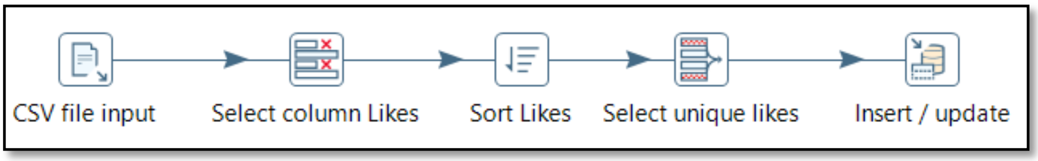
It should be noted that for the zones and products dimensions, separate transformations extracted the data directly from the initial (advertisement) dataset. The reasoning behind this is explained in the Data Mapping section of this document.



3 Data Mapping Process

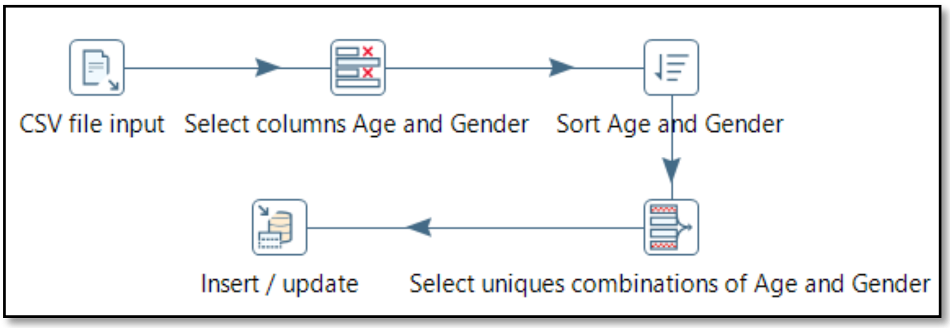
Using the prepared CSV from the extraction process, the following dimensions were mapped using a separate transformation for each of the six dimensions:

DIM_LIKE (TR_ DIM_LIKES, 6.9 seconds to finalise): We mapped out each of the 8 unique likes into the likes dimension using a 5 step process. The first step was a data input of the CSV, after which we selected only the likes column (likes_name) followed by a sorting of the likes and extraction of the unique rows (the latter two as required by Pentaho). The unique likes were then inserted directly into the respective dimension using an insert/update step.

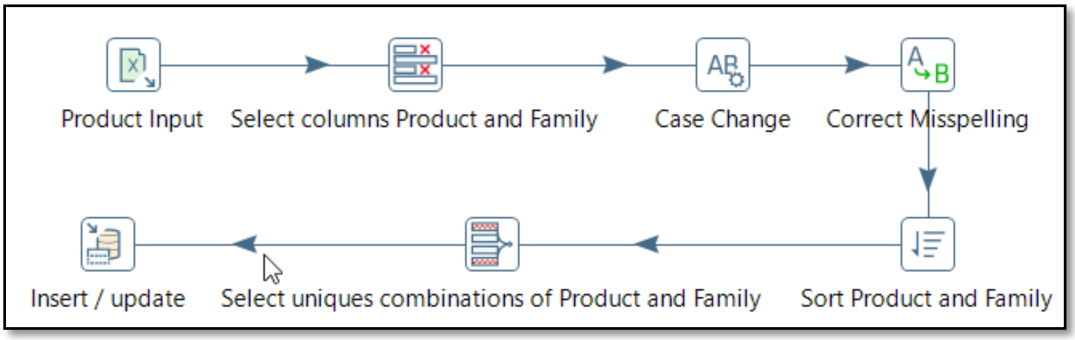


DIM_PLATFORM (TR_DIM_PLATFORM, 8.8 seconds to finalize): The mapping for each of the 4 unique platforms into the platform dimension involved the same steps as for the likes dimension. The first step was a data input of the CSV, after which we selected only the platform column followed by a sorting of the platforms and extraction of the unique rows (the latter two as required by Pentaho).

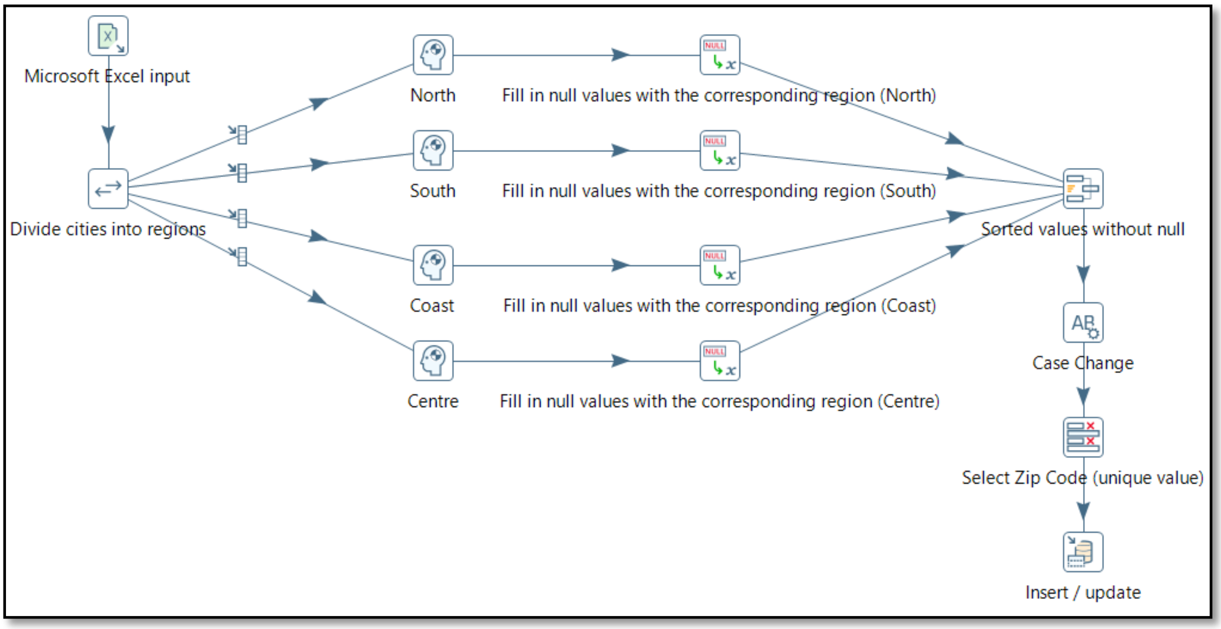
DIM_GENDER (TR_DIM_GENDER, 7.7 seconds to finalize): The gender dimension mapping process was also identical to that for the like dimension. The first step was a data input of the CSV, after which we selected only the target_gender and target_age_range attributes followed by a sorting and extraction of the unique rows. Since our dataset contained two gender groups (Male and Female) and four age groups (18-30, 31-45,46-60, and 61-99), 8 unique combinations were inserted directly into the respective dimension using an insert/update step.


CSV file input
Select columns Age and Gender
Sort Age and Gender
Insert / update
Select uniques combinations of Age and Gender

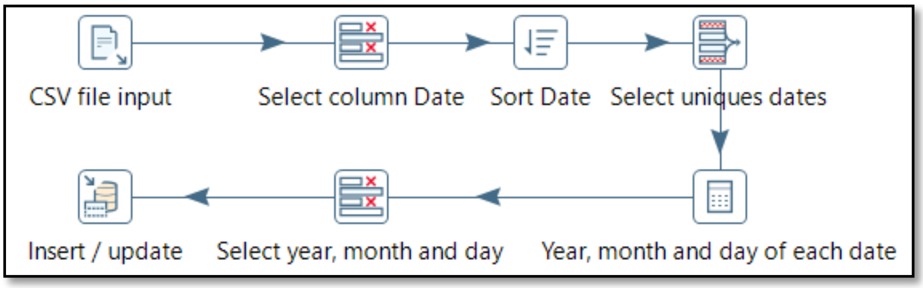
DIM_PRODUCT (TR_DIM_PRODUCT, 0.1 seconds to finalize): The product dimension was mapped using the initial (advertisement) dataset instead of the CSV file from the extraction process. This was done because the PRODUCTS sheet on the initial dataset contained the family names and the product associated to each family while the “PLATFORM” datasets contained only the product names. In other words, product names were associated to the print_value (Prints) and hit_value (Hits), but the family names were not. Thus, instead of adding the family names for each of the products into the clean dataset during the extraction process, which would have made the CSV dataset larger and possibly taken up more processing time, we decided to use the product_name attribute as a means to relate the print_value and hit_value to the family_name. The downside of optimizing the processing time was that the Case Change and Correct Misspelling steps (only for the fields related to the products dimension) from the data extraction process had to be repeated to ensure consistency across the dimensions.


Product Input
Select columns Product and Family
Case Change
Correct Misspelling
Insert / update
Select uniques combinations of Product and Family
Sort Product and Family

DIM_LOCATION (0.7 seconds to finalize): The location dimension was also mapped using the initial (advertisement) dataset instead of the CSV file from the extraction process for the same reason as that for the product dimension. In this case, the ZONES sheet on the initial dataset contained the zip code, zone and city names while the “PLATFORM” datasets contained only the zip codes. In other words, zip codes were associated to the print_value and hit_value, but the city and zone names were not. Thus, instead of adding the city and zone names for each of the products into the clean dataset during the extraction process, which would have made the CSV dataset larger and possibly taken up more processing time, we decided to use the zip_code attribute as a means to relate the print_value and hit_value to the city_name and zone_name attributes. The downside of optimizing the processing time was that we had to first associate the missing zone names from the initial dataset to each of the zip codes before inserting 16 unique zip-codes and their associated city and zone names into the location dimension.

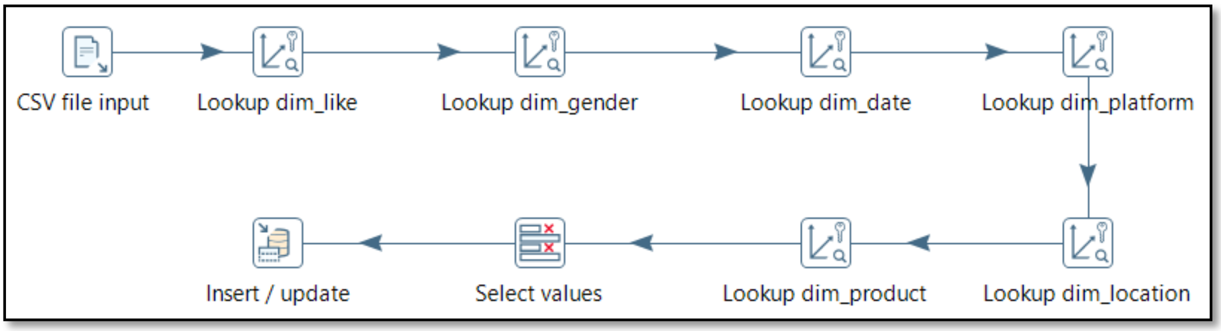

Microsoft Excel input
Divide cities into regions
North
South
Coast
Centre
Fill in null values with the corresponding region (North)
Fill in null values with the corresponding region (South)
Fill in null values with the corresponding region (Coast)
Fill in null values with the corresponding region (Centre)
Sorted values without null
Case Change
Select Zip Code (unique value)
Insert / update

DIM_DATE (6.9 seconds to finalize): Since the data extraction step split the date in the initial dataset into day, month and year, the mapping of this dimension also divided the initial date into the three categories. We tried to extract the unique day, month and year values from from CSV files and map that into the date dimension, but by doing so we could either get 31 unique days or 12 unique months or 1 unique year instead of 365 combinations of days, months and year, which is what we needed for our dataset.

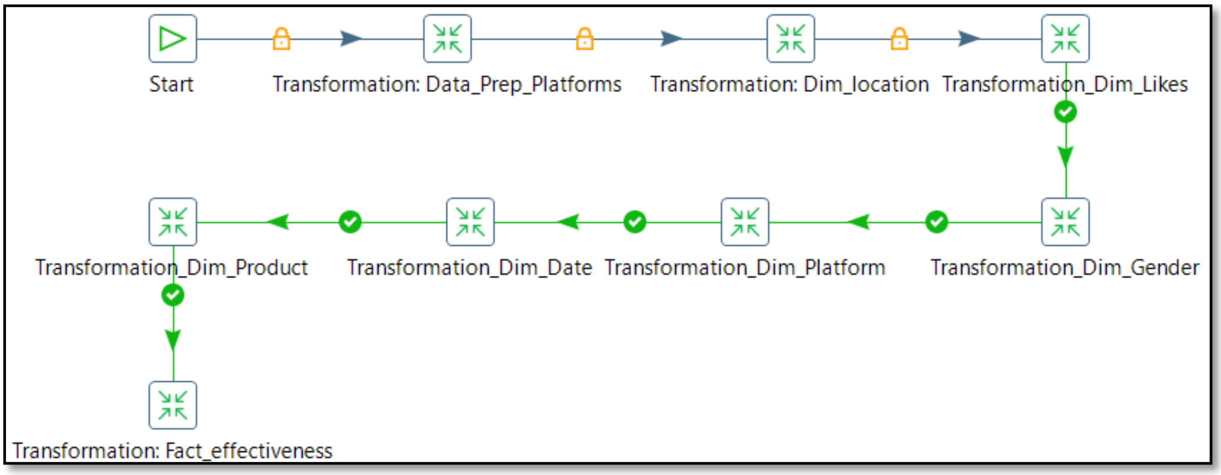


We note that the primary keys for each of the dimensions were automatically created as a part of the modelling process in MySQL. Once the relevant dimensions were mapped, during which the primary keys were automatically created, a separate transformation was created to map the fact table using the relevant foreign keys. The steps were as follows:

FACT_EFFECTIVENESS (seconds to finalize): 6 lookup steps were used find the primary keys for each of the 6 dimensions. After the lookup sequence, the foreign keys as well as the remaining measures (print_value and hit_value) were inserted into the fact table.



Once the transformations for each of the six dimensions and fact table were created, a job (JOB_INTEGRATION) was created to sequence each of the transformations and carry out the entire ETL process.


Start
Transformation: Data_Prep_Platforms
Transformation: Dim_location
Transformation_Dim_Likes
Transformation_Dim_Gender
Transformation_Dim_Platform
Transformation_Dim_Date
Transformation_Dim_Product
Transformation: Fact_effectiveness

4 Data Quality Tracking & Metadata Approach

The majority of steps/approach for data quality tracking and metadata have been described throughout the document (e.g Case Change).