|  |
| --- |
|  |

Talend Project

**Prepared by**

Rohan Umakant Sawant

This a report prepared for understanding the process and approach that was taken to design a database system for a Sales Transaction application with Talend and MySql

Contents

[A. Target Database ER diagram: 2](#_Toc482751831)

[B. Data Understanding 3](#_Toc482751832)

[a. Useless Columns: 3](#_Toc482751833)

[b. Values Identified for check constraints and fixes that needed to be made: 3](#_Toc482751834)

[c. Tables that appear to represent the same entity. 4](#_Toc482751835)

[C. Talend Jobs 5](#_Toc482751836)

# A. Target Database ER diagram:

This ER diagram was made by following the rules of normalization and derived from the data from: <https://github.com/Rohan9969/Talend-Project/tree/master/Data>



Figure 1: ER Diagram

# B. Data Understanding

### Useless Columns:

All these columns were dropped because the data seemed useless and could have been derived again

|  |  |
| --- | --- |
| **TABLE NAME** | **COLUMN NAME** |
| Channels.csv | CHANNEL\_TOTAL |
| Customer\_ext.csv | CUST\_TOTAL |
| Customer int.csv | COUNTRY\_TOTAL |
| products.csv | PROD\_STATUS, PROD\_TOTAL |
| promotions.csv | PROMO\_TOTAL |

Table 1 Useless columns

### Values Identified for check constraints and fixes that needed to be made:

1. For CUST\_GENDER the distinct values are ‘M’ and ‘F’
2. For CUST\_MARITAL\_STATUS the distinct values are ‘-‘, 'Separ.', 'NeverM', 'Mabsent', 'Widowed', 'single', 'married', 'Divorc.', 'widow', 'Married', 'divorced' and 'Mar-AF'.

In CUST\_MARITAL\_STATUS we see a lot of overlapping values which will be fixed as follows:

* -, 'Mabsent', 'Mar-AF' 🡪 ‘Not sure’
* 'Separ’, 'Divorc’, 'divorced' 🡪 ‘Divorced’
* 'NeverM', 'single' 🡪 ‘Single’
* 'Widowed', 'widow' 🡪 ‘Widowed’
* 'married', 'Married' 🡪 ‘Married’

Thus, the distinct values are ‘Not sure’, ‘Divorced’, ‘Single’, ‘Widowed’ and ‘Married’

1. ‘ad news’ in PROMO\_CATEGORY seems to be a subcategory. Hence, we will remove ‘ad news’ from PROMO\_CATEGORY.
2. For PROD\_CATEGORY for ‘ELECTRONICS’ we see there is only one PROD\_SUBCATEGORY which is ‘Photo’. Hence we will replace ‘ELECTRONICS’ by ‘Photo.

### Tables that appear to represent the same entity.

1. If we look at csv’s ‘Customer int.csv’ and ‘Customer\_ext.csv’ they appear to represent the same entity
2. ‘LI\_SALES\_12\_13.csv’ and ‘LI\_SALES\_14.csv’ also appear to represent the same entity

# C. Talend Jobs

The below link has the Talend project code: <https://github.com/Rohan9969/Talend-Project/tree/master/SALESTRANSACTION>

The process of loading target database included 2 steps. First all the data was moved from csv’s to STAGE database in MySQL with no changes in structure of data except for a few. Then the data was moved from STAGE to TARGET database with transformations. All the movement of data was done using Talend Studio.

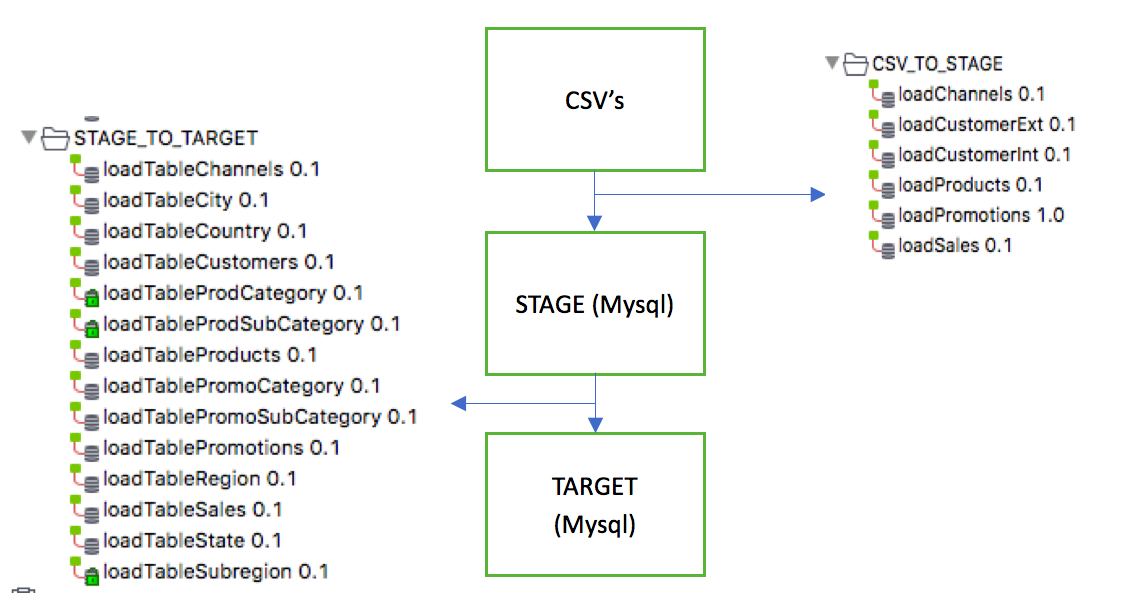
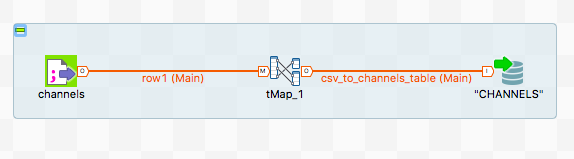


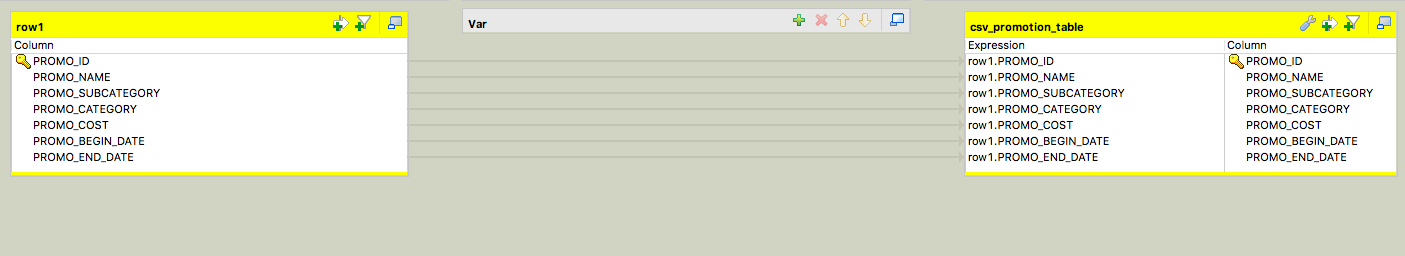
Figure 2: Flow of Talend jobs

Below are sample Talend Jobs:

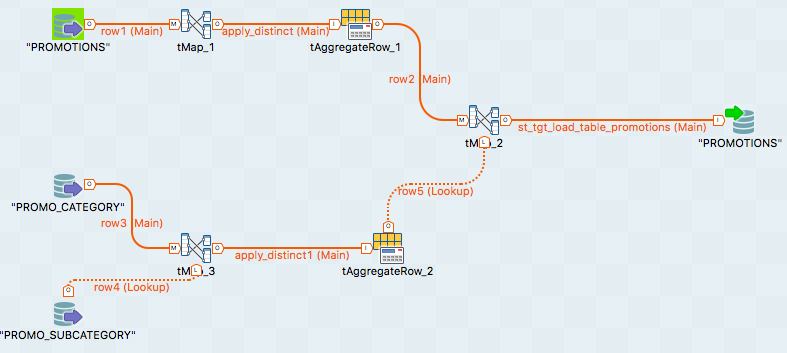
1. CSV to STAGE: Most of the csv to STAGE jobs were nearly the same.Below is showing a simple loading of csv to channels



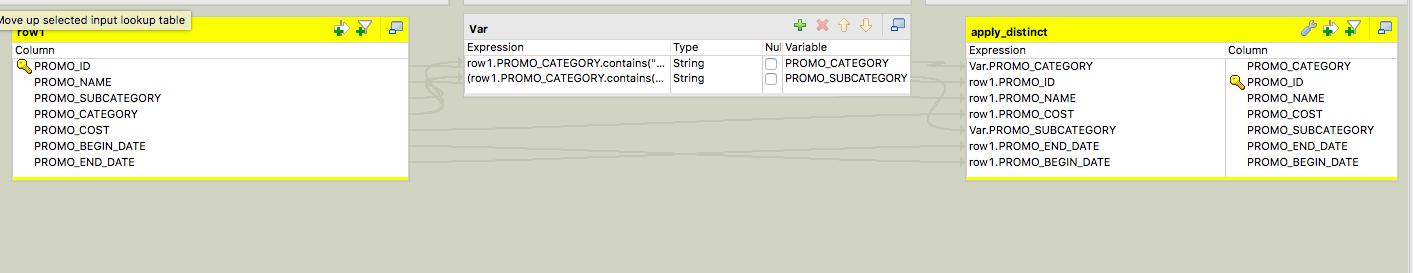
Inside the mapping:



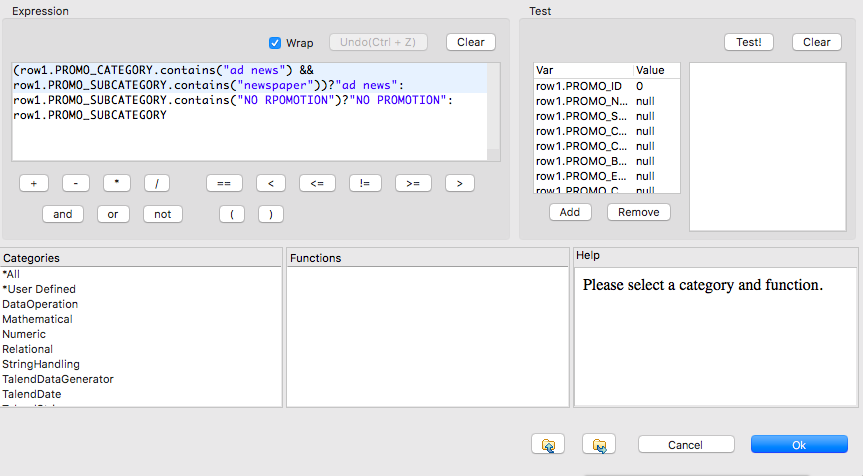
1. STAGE to TARGET: Below is a sample showing the loading of PROMOTIONS table. The aggregate was used to put distinct on the columns.



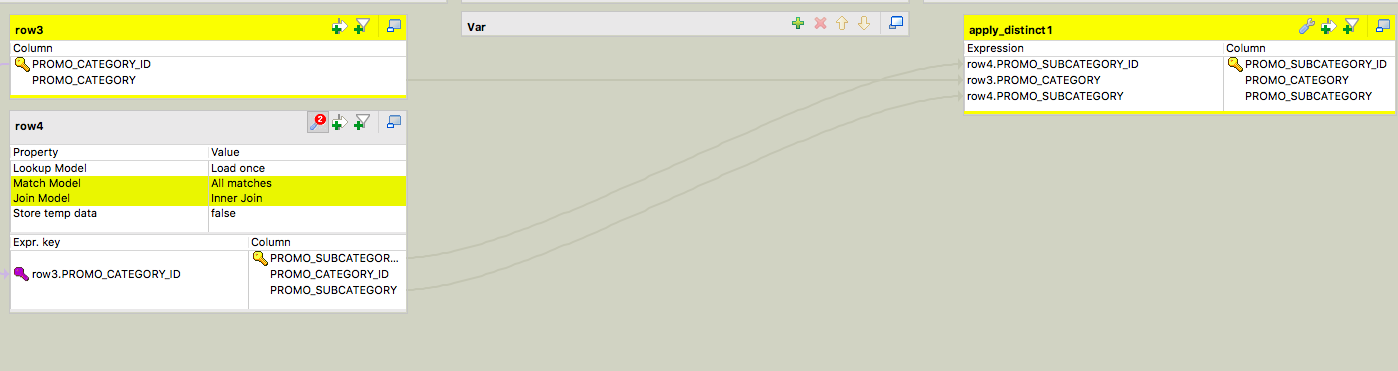
Inside the mapping:



This one required us to calculate some values for transformations.



Even some joins were required which were done as follows:



Most of the tables were designed with the above style. For more snaps of jobs please check the link: <https://github.com/Rohan9969/Talend-Project/blob/master/Documents/Talend%20Jobs.docx>

For MySQL tables created please check the link: <https://github.com/Rohan9969/Talend-Project/blob/master/Documents/mysql_tables.txt>