

Project

Topic:

The idea of this project is to provide businesses with detailed monthly reports about their performance.

Tools:

Talend to transform my data, and **Power BI** to load and visualize it.

Data sources:

My data sources are divided into three types:

1. Excel:

- 'Time': details related to the delivery time of each order

2. CSV:

- 'Commandes': Main information about orders
- 'Articles': Detailed data about each specific article sold

3. JSON:

- 'Reclamations': Claims received from clients

Main Steps:

I. Data Cleaning:

As a first step, I checked my data sources. I made sure that my data is consistent, complete, and relevant.

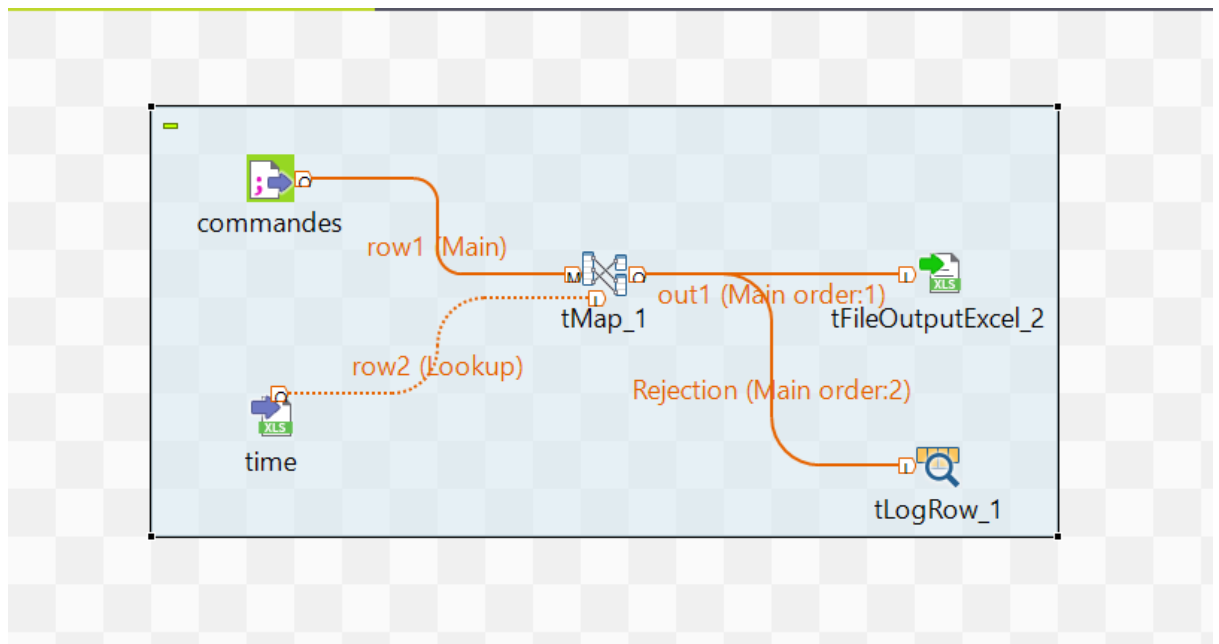
II. Talend: I prepared 3 main jobs:

N.B: when I downloaded my file sources, I set '0' as the default value (for columns related to prices/revenues). So that I won't have any null value.

1. First job (Revenuesvsdate): a job where I used tMap to:

- Link data about orders ('Commandes' file) with details about timing ('time' file)

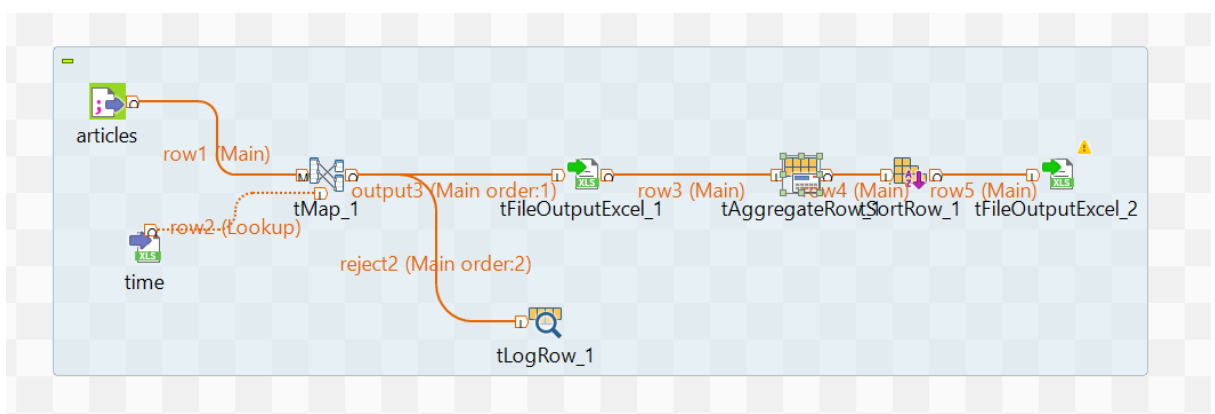
- Filter only the orders that weren't canceled (code: `row1.Annulee_.equals("Non")`)



=> we got an output Excel file with the filtered data

2. Second job (part3):

- I used tMap to link data about articles ('Articles' file) with details about timing ('time' file)
- Filter the orders that weren't canceled
=> we got an output Excel file 'Part 3'
- Grouped the number of orders per category and per order
- Sorted the output file in ascending way (Sorted by Category and Article)
=> we got an output Excel file 'Aggregation out3'



3. Third job (part4):

Schéma Built-In Modifier le schéma ... Sync colonnes

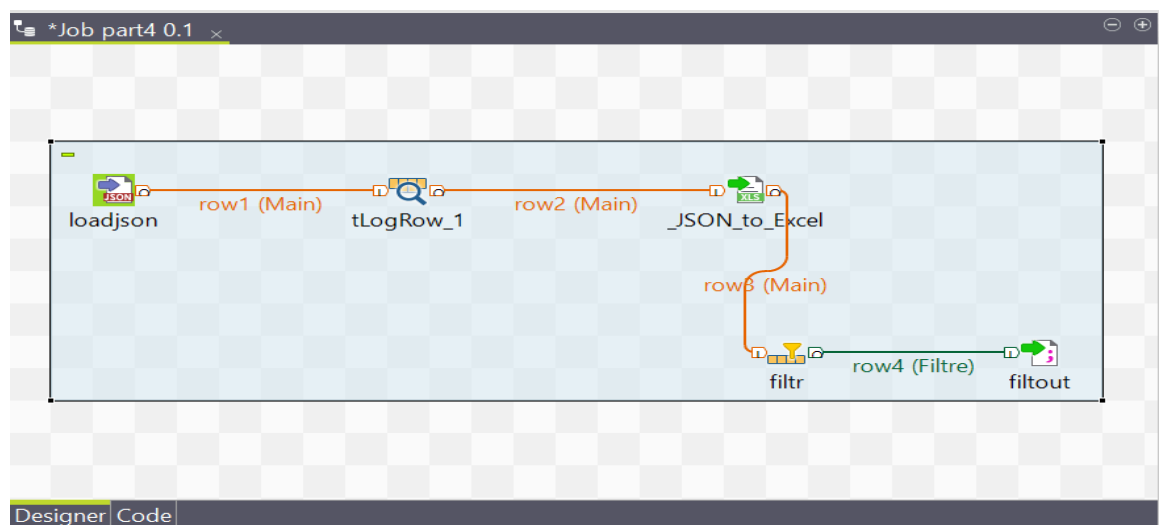
Group by

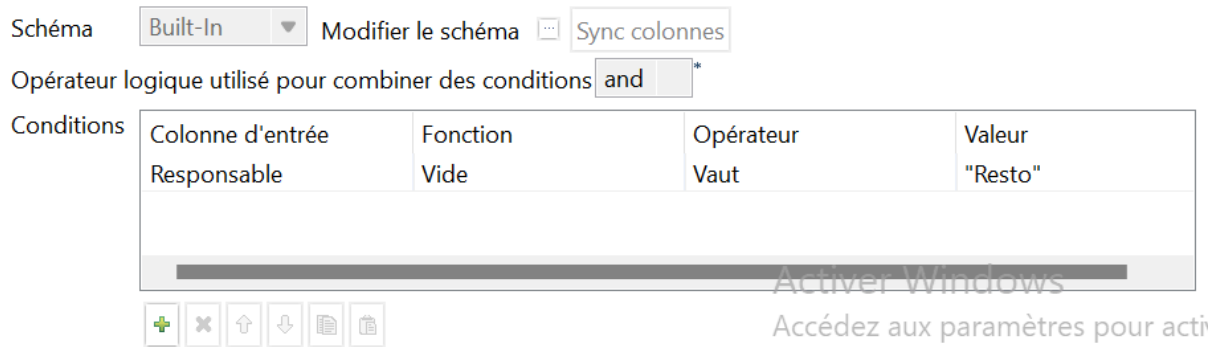
Colonne de sortie	Position de la colonne d'entrée
Categorie	Categorie
Article	Article

+ × ↑ ↓ 📄 📋 +

Activer Windows

- I transformed my JSON file into an output Excel file
- I filtered the data to get only the claims that are due to a restaurant's fault not the startup's,.
=> we got an output CSV file 'Part 4'





=> Talend is my **staging area** where I extracted my data and then transformed it.

III. Power BI:

A. Data Warehouse part:

I loaded my data on my **Power BI Desktop**:

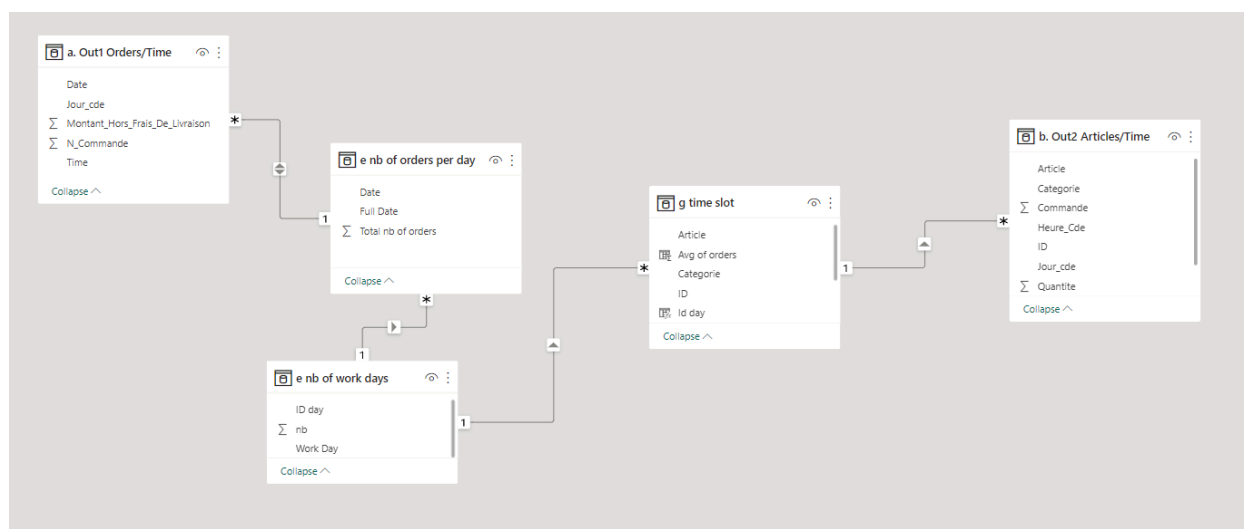
1. I created three new tables

- I grouped the 'total number of orders per date'
- I grouped from the new table the 'number of work days per weekday'
- I grouped data (from the 2nd Talend output table) per category => article => weekday => Start Hour

Then I used **Power Query Formula Language** ('M' language) to import the nb of work days per weekday (used the **VLOOKUP** function)

Finally, I added new columns to count the average of articles sold (used the **DIVIDE** function)

2. Data Modeling: I created the relationships between my facts and dimensions tables.



- => 'time slot' is my principal fact table that I created from two dimensions
- => 'nb of orders per day' is an aggregation table I made from Out1
- => 'nb of work days' is an aggregation table I created from 'nb of orders per day'

3. I renamed my tables and columns in a user-friendly way
4. I created new columns: e.g Id day (I needed this column to sort weekdays in my matrices from Monday to Sunday and not in alphabetical order)

B. Data Visualization:

- I. I created a **report** that includes 7 pages:

1. **Orders vs dates**

- We can notice that the highest revenues were on the 18th and the lowest were on the 16th: the manager should investigate the reasons for such a decrease
- No revenues between the 9th and the 10th and that's possibly because of Aïd Al-Adha

2. **Orders per weekday**

- The highest nb of orders was on Monday: The manager should focus on this weekday to avoid any delay or mistake while preparing the orders

3. **Revenues per category**

- The category with the highest revenues were 'BURGER'
- The lowest percentage of revenues was for 'Pizza' => this category did generate only 0.11% of the restaurant's revenues: it's important to find solution (whether to introduce new recipes/ discounts / eliminate it from the menu..)

4. **Orders per category**

- Over 50% of the orders were for 'Burgers' => They should make sure that enough ingredients of this category are available to meet the high demand
- Very low demand for 'Pizza' (0.05%) => this confirms that a solution should be implemented to attract more people for this category

5. **Articles revenues per Category**

- For Burger: Cheese Bum generated the highest revenues.
- For Desserts: Chocobout generated over 60% of the revenues

- For 'Our best sellers': some articles like Cheesy dog or Smoky burger porc generated around 0% of the revenues

6. Articles orders per Category

- For Desserts: Lowest demand was for Cheescake Chocolat (around 1.5%)
- For Burger: Cheese Bum had the highest revenues. We can take advantage of these articles to promote others (e.g introduce menus with Cheese Bum and Cheescake chocolate with a small discount)
- For 'Our best sellers': some articles like Cheesy dog or Smoky burger porc had around 0% demand => the restaurant should consider the option of eliminating such articles from this category (as they are not best-sellers)

7. Time Slot

- On Monday: the restaurant received the highest number of orders between 12-2 pm and 7-10 pm. Starting from 8 pm it was over 30 orders per hour => The manager should keep the option of increasing the nb of his staff members during these periods
- During the weekend (Th-Su) the pick was between 7 pm-10 pm

8. Claims

- Number of claims was 18 with 9 claims because of 'Missed Articles': it's important that employees check the order before they deliver it
- This restaurant had the highest number of claims compared to the others => The manager should set a strategy to avoid such a number in the future (Possible solution: add a staff member who should verify the nb and the quality of the articles before it gets delivered)

II. I did **publish** the report and created a **dashboard** on the Power BI service + I did pin **three cards** that highlight total revenues + total nb of orders during the month + nb of categories the business has.

!! The report & the dashboard are both submitted with this document !!