BBG CASE STUDY

Data and source files:

campaign_object.json

amazon_shop_mapping.csv

sales_and_traffic_data.csv

Currency conversion API example:

example: https://currencylayer.com/documentation

PS: to be able to use this api, you need to sign up and create an account (free of charge)

Case overview

In an e-commerce company, daily challenges for a data team are the norm. One of these challenges is to daily get the latest cost and sales data from external apis, process the data, transform it and move it to our DWH to be available for BI Team and business partners for daily analysis and steering.

During this exercise we will present some of the daily challenges that we face.

DATA DESCRIPTION :

- **sales_and_traffic_data.csv** : representing the sales data for items sold on amazon during a period of time along with traffic data such as sessions and page views.
    - containing 13 columns:
        - child_asin: is the amazon product id (Amazon Standard Identification Number)
        - sessions : number of sessions for a product in that day
        - page_views : number of page views for a product in that day
        - units_ordered : number of units ordered for a product in that day
        - units_ordered_b2b : number of units ordered from another business for a product in that day
        - ordered_products_sale : sales made from a specific product in a certain day (amazon shop currency)
        - ordered_products_sales_b2b : sales made from a specific product in a certain day (amazon shop currency)
        - total_ordered_items : total ordered items
        - total_ordered_items_b2b : total ordered items b2b
        - region : country code
        - shop_name : shop name in amazon
        - report_date : date of the report
- **amazon_shop_mapping.csv** : a mapping file between amazon shops and our internal DB
    - containing 4 columns:
        - shop_name: representing the name of the shop in amazon
        - country : representing the country code
        - id : the shop id and definition in one of our databases
        - currency : shop currency in amazon
- **campaign_object.json** :

- containing 4 columns ;
    - STARTDATE : represents the start date of the ad campaign
    - CAMPAIGNID : represent the ad campaign name
    - CREATIVE : containing a dictionary describing the brand in the campaign (brandName, brandLogoUrl, asins included in the campaign from that brand, etc)

Case 1.1.

Before the data is available for business partners and BI teams, it should pass by multiple transformations and processing to ensure that the final data is clean and usable.

For example, all the sales data in **sales_and_traffic_data.csv** is in different currencies (currency set in the shop creation in amazon), therefore we must convert all the sales columns to euro to enable aggregate reports to include all shops in the final dashboards. Unfortunately, amazon api does not include the currency of every sale made in its reports, therefore we made the file **amazon_shop_mapping.csv** containing the currency of every shop in amazon.

Questions:

please use Python and Pyspark only:

1. Convert the sales data from **sales_and_traffic_data.csv** to euro by using **amazon_shop_mapping.csv** and the currency conversion api presented or any forex api from your choice (python packages not allowed).
2. Calculate the total revenue.
3. In separate Dataframes present the total revenue per country and per shop and per month.
4. Replicate the above questions in SQL.

Case 1.2.

campaign_object.csv is a snippet of another api response we get from amazon api, containing information about our active and inactive ad campaigns.

One column in this response, named “CREATIVE”, is an object containing information about the brand, which is included in the campaign, and the asins (Amazon Standard Identification Number), which are pushed in the ad within this brand. The maximum number of asins within this column is 3.

example : `'asins': ['B06W5543D6', 'B074F576VM', 'B078VLNF5K']`

Questions:

please use Pyspark and SQL :

1. Transform the **campaign_object.csv** by separating the column CREATIVE to multiple columns (`brandName` , `brandLogoAssetID`, `headline`, `asins`, `brandLogoUrl`).
2. Separate the new formatted column `asins` to 3 new columns `asin_1` ,`asin_2`, `asin_2`
3. By using **sales_and_traffic_data.csv,** extract the distinct asin list and save it into a dataframe.
4. Create a new column in the new dataframe from **campaign_object.csv,** called `active_asin`, containing the first asin that exists in the formatted distinct asin list.
    - Example: if `asin_1` exist in the distinct asin list then `active_asin` = `asin_1` otherwise a comparison with `asin_2` is needed and if `asin_2` does not exist in the distinct asin list, we move to `asin_3` comparison.

Please write your solution in a jupyter notebook where all the steps are well presented and commented.