Intern - Data Analytics & Strategy at Mamaearth Task

Task 2

I would use an SQL database to store this data because

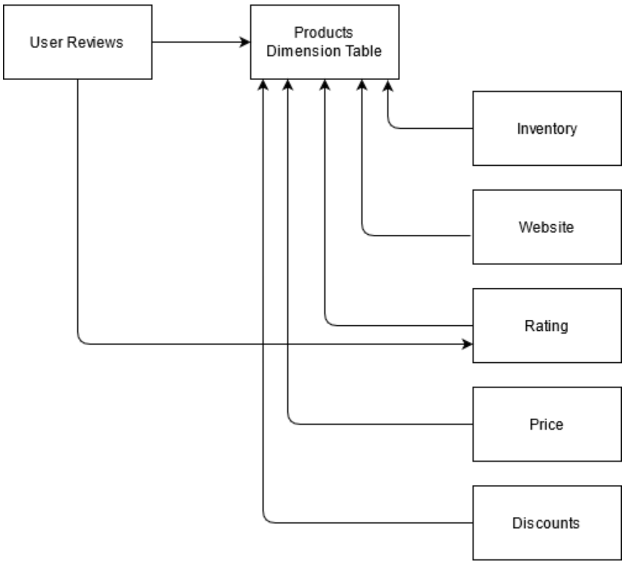
- The structure of our data is unlikely to be changed frequently, hence a pre-defined schema can be used
- A Redshift like DBMS would solve for setting up ELT pipelines, and making integration easier with dynamic in-flowing sales data
- Widely accepted
- Allows analytics both on SQL as well as direct integration with python, R etc.

As this data needs to be updated on a daily basis, we would need to separate the data fields to allow for time series analysis as well.

A “created at” and “updated at” field for every insertion would allow to keep track of changes as well as review them over time.

I would structure the data into tables as illustrated below:

- Product Dimension Table: MPN (Primary Key), name, pack_size, ingredients, category, subcategory
- Inventory: MPN, date, stock flag
- Rating: MPN, date, aggregate rating, review count
- Website: MPN, date, image link, url, position
- Price: MPN, date, MRP, currency, valid till
- Discounts: MPN, date, discounted_price, discount, valid_till
- User Reviews: MPN, review, rating, author, date published



Hence to summarise, I would be using 7 tables to store the scraped/available data.

Additionally, there should be a user id or review id to uniquely identify a review and identify the user that wrote the review.

To ensure integration with any future data (user level. Sales data etc.), we would ensure the connections between a product <> sales <> user <> reviews are present.

To automate this process, python can be used to extract the data, load it in an intermediary like CSV raw data files to allow for transformation required if any, or directly through pandas dataframe.

Alternatively, these jobs can also be set up with the help of platforms like AWS, Kafka or Airflow. The choice largely depends on the quantum of data and the frequency of refresh, keeping the cost as a constraint metric.