Data Modeling documentation - Yalo Case

Business Overview

Local retail company focused in beverage sales, localized in Iowa-US, which wants to use data analytics to understand financial performance and get insights for next actions, for both finance and sales department (main stakeholders)

Tech Stack

Data Source

Considering a paid service that is an API Endpoint with daily information about all liquor sales in Iowa.

Tools

Considering a rather small company, the data stack will follow a full open source scenario

- Linux Server
- Python
- PostgreSQL
- dbt
- Airflow
- Power BI / Looker Studio

Ingestion Phase

- Python Script + Airflow

A python script that will use a GET request into the API to gather the **previous day** data, every morning at 8AM.
To avoid risks of losing data or any eventual issue in the script, the code will check the last updated date on the PostgreSQL bronze layer and will get any day that is not present. So, in case of a failed run, it may get the missing days as well
Everything orchestrated by airflow.

At the end of the code, the python script will return a .parquet file for each day in a folder, that would be the raw layer

Data Modeling

Bronze Layer

The bronze layer will have a single table called *t_bronze_iowa_sales*, that would be inserted by a Python script in PostgreSQL database db_bronze_iowa.
The ingestion type would be **incremental**. Only adding new days into the *t_bronze_iowa_sales* .

At the beginning of dbt model run, it will perform quality tests into this table to check

- Duplicity
- NULL values

Stopping the pipeline in case of any of these issues appears, sending a mail/message to the team to verify what happened.

This table will only be accessed by Data & Analytics engineering team.

t_bronze_iowa_sales	
invoice_and_item_number	*TEXT*
date	*DATE*
store_number	*TEXT*
store_name	*TEXT*
address	*TEXT*
city	*TEXT*
zip_code	*TEXT*
store_location	*TEXT*
county_number	*TEXT*
county	*TEXT*
category	*TEXT*
category_name	*TEXT*
vendor_name	*TEXT*
vendor_number	*TEXT*
item_number	*TEXT*
item_description	*TEXT*
pack	*INTEGER*
bottle_volume_ml	*INTEGER*
state_bottle_cost	*NUMERIC(10,2)*
state_bottle_retail	*NUMERIC(10,2)*
bottles_sold	*INTEGER*
sale_dollars	*NUMERIC(12,2)*
volume_sold_liters	*NUMERIC(12,2)*
volume_sold_gallons	*NUMERIC(12,2)*

Silver Layer

Here we will apply the Star Schema methodology to model this data, based on the business needs:

fact Table: t_fact_sales

- A materialized table with informations about sales, in a daily granularity

dimension tables:
separated by domain

- dates: t_dim_dates
  - with information about weekday, year, quarter, etc
- stores: t_dim_stores
  - detailed view of stores and its locations
- products: t_dim_products
  - details such as pack size, volume per bottle, etc
- vendors: t_dim_vendors
  - vendor names

Everything would be orchestrated by dbt + Airflow, with a daily model run.
At the end of each run, there will be tests as well.

Gold Layer

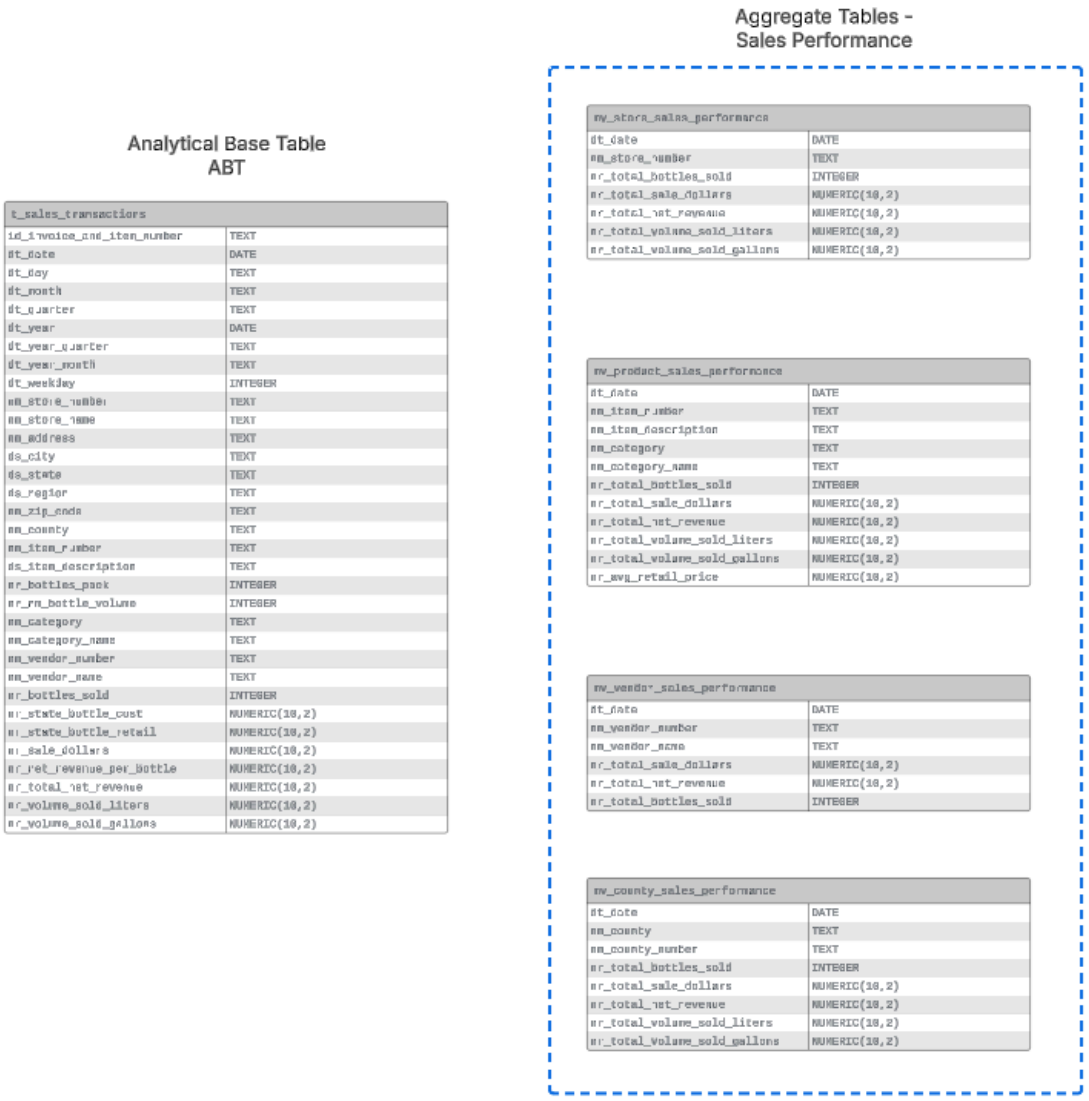
Here the modeling is focused in 2 approaches

ABT (Analytical Base Table)

A single source of truth for BI teams and Data Scientists, with tested and clean data to link to a BI tool (Looker Studio or Power BI). And for Data Scientists, allowing them to prepare Machine Learning Models (such as Time Series forecasting). In case of a MLOps team, this would be the beginning of their workflow.

Performance Tables (Materialized Views)

Aggregated materialized views, which could provide status of sales per vendor, product, store and county, allowing business teams to have fast answers and also use it on BI tools, as shown in the chart below



At the end of silver layer processing, there will be several tests before creating those tables and materialized views, with actions in case of trigger

- Not null - Stop flow
- Duplicates - Stop flow
- Sales standard deviation - Warning
- Outliers - Warning

Overall Architecture view: