Data Warehouses

Hugo Valdrez
Computer Science Master's Degree
Universidade do Porto
up201704962@up.pt

João Carvalho
Data Science and Engineering Master's Degree
Universidade do Porto
up201507023@up.pt

Luís Pinto
Computer Science Master's Degree
Universidade do Porto
up201704025@up.pt

Bruno Silva
Data Science and Engineering Master's Degree
Universidade do Porto
up201508756@up.pt

***Abstract*—In this report, we use a raw dataset with information on hundreds of thousands of sales of cigarettes and related products in several countries of Africa to design a data warehouse. After understanding the properties of the dataset we idealize the structure of the data warehouse through a bus matrix and after, a dimensional model. Through an ETL process we extract and transform the dataset into a format that is more usable and then load the dataset into the data warehouse taking into consideration our design. With all the data loaded, we then are able to query the data warehouse and perform some relevant data analysis.**

IN an increasingly data-driven world, the ability to effectively manage, analyze, and derive insights from vast datasets is of extremely relevance, being particularly crucial on Business Intelligence. In this assignment, we propose a structured design for a data warehouse tailored specifically for the cigarettes market in Africa [1], encompassing both sales transactions, vendors' purchases, and stock management.

Our primary objective was to construct a comprehensive data warehousing solution that not only centralizes data but also facilitates efficient querying and analysis. To display and detail our entire work, we organized this report with the following structure:

Section I, Groundwork for our data warehouse design. This involved planning and creating the dimensional bus matrix that serves as the basis for our data warehouse to outline the various dimensions and facts relevant to our dataset. Additionally, we present a dimensions and facts dictionary to provide a clear understanding of the data elements and their relationships.

Section II, Construction of the dimensional data model. This model representation was essentially the blueprint for organizing and structuring our data warehouse, ensuring the desired ease of analysis that characterizes a data warehouse as well as querying performance.

Section III, Data selection process explanation through the lens of an ETL (Extraction, Transformation, and Loading) framework. We identify and describe the data sources for our project, go through the whole set of data transformation techniques we used and the reasoning behind our decisions, and finally show the process of data loading into our database system. The main objective of this section is to show how the data is carefully structured and treated from a raw format to the final one which we can comfortably load and query.

Section IV, Querying. Once our data warehouse was populated with relevant data our objective was to learn interesting insights regarding the subject of the cigarettes market in Africa. This involved exploration of the results through querying.

Section V, Data Analysis. We complement our analysis with Excel as our data visualization tool, making use of PowerQuery and DAX. With it, we created informative dashboards to effectively present and interpret our findings.

Section VI, Advantages and disadvantages of our data warehousing solution, and data warehouses in general, in comparison to traditional operational databases. We distinguish both approaches from the theoretical standpoint and what they bring to the target users that will need to use the data or deal with the database system itself.

Section VII, Key takeaways, insights gained, and lessons learned throughout the process of designing and implementing our data warehouse. We also propose different things that could be done to enrich our design even further.

Data Preparation Note: Prior to the construction of our data models, the Kaggle dataset underwent a series of transformations to better align with our analytical objectives:

1. Dataset Customization:

- Column Management: Removal of extraneous columns to streamline the dataset.
- Identifier Modification: Adjustment of identifiers to ensure data integrity and consistency.
- Data Point Standardization: Renaming and standardizing data points for uniformity across the dataset.

2. Data Enrichment: To enhance our data model, we:

- Simulated Purchases Table: Generated a randomized purchases table based on sales data, which allowed us to introduce additional fact tables.
- Stocks ETL Process: Established a robust ETL process that harmonizes sales and purchases data, culminating in the creation of a comprehensive stocks fact table.

I. Data Warehouse Design

In an initial analysis of the dataset we used, there were 28 columns. While not all being deemed useful, certain columns stood out, allowing us to determine what was considered as dimensions or facts.

Dimensions:

- DIM_STORE: This dimension provides information for each store, including details such as store type and subtype.
- DIM_LOCATION: This dimension offers detailed information regarding the store's location, down to the suburb level granularity. Additionally, it incorporates hierarchical structures, encompassing suburb, province, city, and country.
- DIM_TIME: This dimension operates at the granularity level of a day, providing details such as weekday, month, and year.
- DIM_PRODUCT: Operating at the product level, this dimension stores information concerning the brand and sub-brand as well as product category. For instance, "Marlboro" represents the brand, while "Gold" denotes the sub-brand, and the product category would be "Cigarettes".
- DIM_Country: Due to the use of external information about population, we needed to use another dimension at a lower granularity for the location since that information is obtained at country level.
- DIM_Year: For the same reason DIM_COUNTRY was created, we needed to establish a dimension for year. DIM_TIME has a granularity that exceeds what is necessary for the population fact.

Facts:

- SALES: This table comprises data detailing which products were sold, along with the time of sale, the store where the transaction occurred, and the corresponding price.
- PURCHASES: This table includes information regarding purchases, specifying the product bought, the store who made the purchase, the purchase price, and the quantity purchased.
- STOCKS: Utilizing sales and purchases data, the stocks table is calculated. This fact table was constructed to track stock levels.
- POPULATION: An external data source was used to obtain the population information for each country of the dataset across time. Here we have information about the total population of each country as well the population size by age group.

Data mart	Star	Dimension	Time	Location	Product	Store	Country	Year
Operations	Sales		x	x	x	x		
	Purchases		x		x	x		
Wares	Stocks		x		x	x		
Demographics	Population						x	x

Fig. 1. Bus Matrix

With the structure displayed in the bus matrix in Figure 1 we are able to cover a significant part of the operation involving the purchasing and resale of various types of products within the cigarette market. We also include information about the inventories available in the stores at a given instant as well as demographic information that can be crossed with the remaining data to extract interesting trends and insights. This demographic information encompasses population statistics for each country, providing valuable context for understanding market behavior and consumption patterns.

II. Dimensional Model

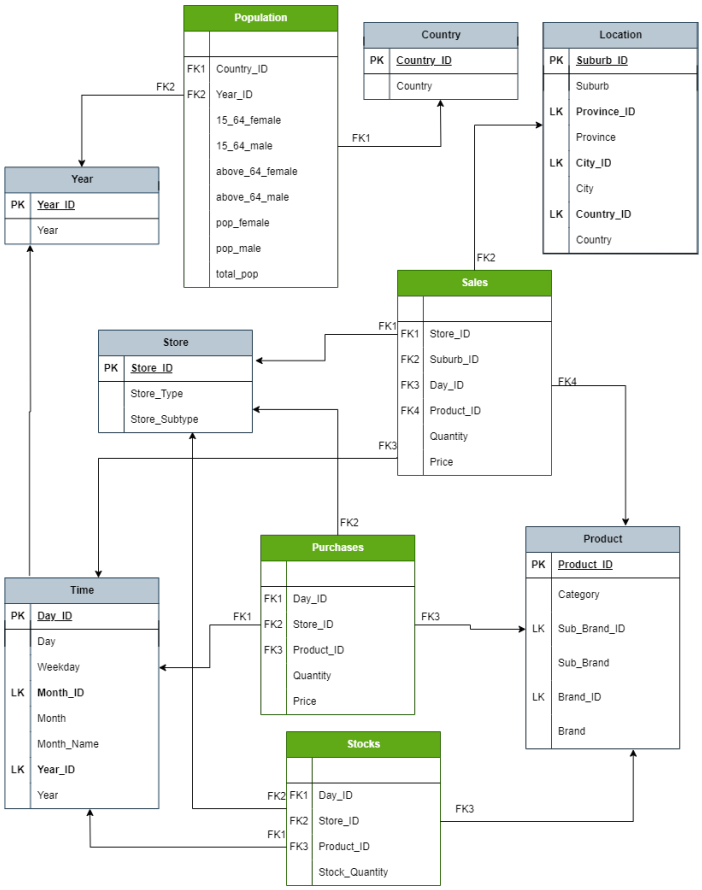

Fig. 2. Dimensional Model

With the dimensions and fact tables defined, along with the completion of the bus matrix, the subsequent step involved constructing the dimensional model. This process results in the data warehouse represented in Figure 2.

The entire data warehouse is composed by four fact tables and six dimensions. In terms of measures, we have included several types. We have additive measures in the form of Quantity and Price inside both the Sales table and the Purchases table. We can call this an additive measure because we can perform different metrics such as totals or averages without the interpretation of the information stopping to make sense. To perform operations like this on semi-additive measures, on the other hand, only makes sense in certain contexts, meaning across only some of the dimensions. For instance, if we take the Stock Quantity in the Stocks table, we can use this measure to get information like the total stock of a given product within a country - across the location dimension - but, at the same time, it would not make sense to add up the stock of a product in a specific store in 2 consecutive days - across the time dimension. The Stock Quantity is the perfect example of a semi-additive measure.

III. Extraction, Transformation, Loading

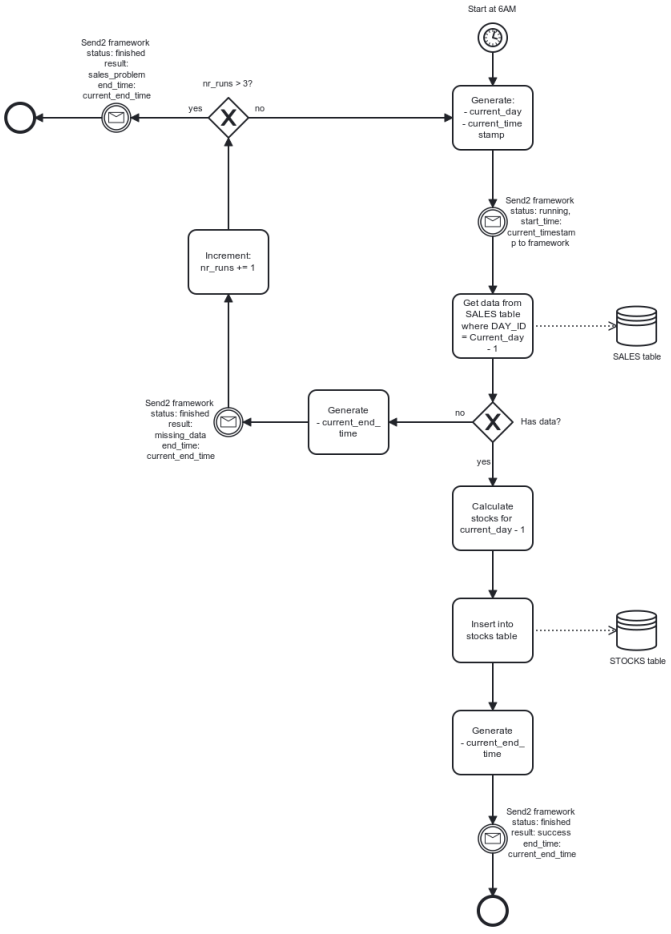

Fig. 3. Stocks Generation BPMN

This section delineates the operational framework of the Stocks Extract, Transform, Load (ETL) process. This process is meticulously orchestrated to commence at a pre-established juncture each day, specifically at the onset of business hours. It operates under the presumption of unfettered access to the sales and purchases data, which are procured from the operational database. These critical datasets are then adeptly utilized within the ETL workflow to accurately populate the stocks table. Furthermore, the delineation of the Stocks ETL process is explained through the utilization of the following Business Process Modeling Notation (BPMN):

As pointed in a previous section, the generation of stock data stands as a pivotal aspect. This process involves the daily aggregation of stock information, with granularity at the day level and coverage of each individual store data. For this, an Extract, Transform and Load (ETL) process has been designed (as illustrated in Figure 3). This ETL process incorporates a framework with a control mechanism, encompassing parameters such as start time, status, end time and result. This would also serve as logging system ensuring a historical record of process execution, enabling identification and resolution of any anomalies or failures encountered during execution. Each morning, at a predetermined hour, the ETL process is initiated and communicates its start to the framework. Subsequently, it retrieves sales data for the current day from the sales table, which is presumed to have been populated by a preceding process. In the event of missing data, the process iteratively attempts retrieval, alerting the framework to the absence of expected data. If the attempts keep proving futile, signaling a potential issue with the sales data availability, an ending event is triggered, documenting both the absence of data and the suspected underlying issue within the sales table preparation process. Upon successful data retrieval, the ETL process proceeds to compute current stock levels, factoring in the previous day's stock figures and recent sales transactions. Finally, the calculated stock data is inserted into the stocks table and the ETL framework is concluded with a successful status indication.

IV. Querying

At this point, we have our data warehouse completely structured and ready for consultation and querying. In order to show the capabilities of a data management system such as this and the easiness with which we can extract information from it, we designed several SQL queries with various approaches, some of which will be detailed in this section.

A. Which country spends the most on cigarettes?

```
SELECT
    Country,
    SUM(Price) as 'Sales Value'
FROM
    dw_cigarettes.sales s
INNER JOIN
    dw_cigarettes.dim_location l
ON
    s.Suburb_ID = l.Suburb_ID
GROUP BY
    Country
ORDER BY
    SUM(Price) Desc;
```

In order to obtain the country with the highest value of sales, we must find the total sales per country using our datasets. This simple query outputs a table detailing the total amount of sales present in our dataset in US Dollar, for comparison purposes. We order the table by this amount so that we have an idea of which countries have more sales in value, in aggregate terms. The output of this query in this case is presented below and for representation purposes we choose to only show the first ten countries.

Country	Sales Value
South Africa	106560.7091
Namibia	86816.8216
Zimbabwe	68279.8210
Botswana	42095.8078
Lesotho	14614.1920
Tanzania	9944.2373
Eswatini	2491.2855
Malawi	1759.1460
Mauritius	1488.0960
Zambia	718.8021

Fig. 4. Partial output of query 1

B. What is the sales trend per country?

```
SELECT
    Country,
    t.Year,
    SUM(Price) AS 'Sales Value'
FROM
    dw_cigarettes.sales s
INNER JOIN
    dw_cigarettes.dim_location l
ON
    s.Suburb_ID = l.Suburb_ID
INNER JOIN
    dw_cigarettes.dim_time t
ON
    s.Day_ID = t.Day_ID
GROUP BY
    Country,
    t.Year
ORDER BY
    Country,
    t.Year;
```

Adding a layer of complexity to the previous question *A*, this question introduced the time dimension into the equation. By joining sales, location, and time we are able to answer it and obtain the total value of sales per country and per year. The partial output of the SQL query is demonstrated below.

Country	Year	Sales Value
Botswana	2016	1346.1169
Botswana	2017	2441.9774
Botswana	2018	8862.9353
Botswana	2019	6443.9268
Botswana	2020	632.0100
Botswana	2021	22368.8414
Cameroon	2021	49.3783
Chad	2019	4.2956
Chad	2020	9.5112
Chad	2021	614.7402
Eswatini	2016	1397.4352
Eswatini	2019	6.1521
Eswatini	2020	0.5100
Eswatini	2021	683.2625
Eswatini	2022	403.9257
Ethiopia	2018	76.2563
Ethiopia	2019	287.6975
Ethiopia	2020	9.1948
Ethiopia	2021	302.9881

Fig. 5. Partial output of query 2

C. How did the cigarette price evolve in each country?

```
SELECT
    Country,
    Year,
    AVG(Price / Quantity) AS 'Average Price'
FROM
    dw_cigarettes.sales s
INNER JOIN
    dw_cigarettes.dim_time t
ON
    s.Day_ID = t.Day_ID
INNER JOIN
    dw_cigarettes.dim_location l
ON
    s.Suburb_ID = l.Suburb_ID
INNER JOIN
    dw_cigarettes.dim_product p
ON
    s.Product_ID = p.Product_ID
WHERE
    Category = 'Cigarettes'
GROUP BY
    Country,
    Year
ORDER BY Country, Year;
```

This is another relevant question to make in order to understand how the cigarette market evolved across the time period of analysis. Here we look into the average price of a single cigarette stick in each country and by year, a similar approach as the one taken to answer question *B*. For comparison purposes, we do not use the local currency price but a standard price in US Dollar for every country. In the table below we once more show a section of the output of the query which clearly evidences not only the average price difference between different countries, but also the price variation in the same country across multiple years.

Country	Year	Average Price
Ethiopia	2018	0.063759448161
Ethiopia	2019	0.049380741735
Ethiopia	2020	0.036029227642
Ethiopia	2021	0.067468329854
Ghana	2017	0.069032187500
Ghana	2018	0.067845545455
Kenya	2018	0.087448386243
Kenya	2019	0.085121062500
Lesotho	2016	0.146659927184
Lesotho	2017	0.155926907413
Lesotho	2018	0.142615756726
Lesotho	2019	0.148138487080
Lesotho	2020	0.125994014002
Lesotho	2021	0.169965225345
Lesotho	2022	0.143672433991

Fig. 6. Partial output of query 3

D. Where is it more expensive to buy cigarettes?

```
SELECT
    Country,
    Store_Subtype,
    AVG(Price / Quantity) AS 'Average Price',
    RANK() OVER (
        PARTITION BY
            Country
        ORDER BY
            AVG(Price / Quantity) DESC
    ) AS 'Price Rank'
FROM
    dw_cigarettes.sales s
INNER JOIN
    dw_cigarettes.dim_store r
ON
    s.Store_ID = r.Store_ID
INNER JOIN
    dw_cigarettes.dim_location l
ON
    s.Suburb_ID = l.Suburb_ID
INNER JOIN
    dw_cigarettes.dim_product p
ON
    s.Product_ID = p.Product_ID
WHERE
    Store_Subtype != 'None'
    AND Store_Subtype != 'Not Applicable'
    AND Category = 'Cigarettes'
GROUP BY
    Store_Subtype,
    Country
ORDER BY
    Country, 'Price Rank';
```

To answer this question we chose to approach it in a more comprehensive way. The idea was to understand in which type of store would be more expensive to buy cigarettes from, and to expand this analysis for every country. To do this we joined every relevant table and calculated the average individual cigarette price per country and store type. To enhance the ease of analysis further, we used a partition by country to create a column with a ranking indicating which store type has the highest average cigarette price for that specific country.

Country	Store_Subtype	Average Price	Price Rank
Mozambique	Petrol Station	0.005755714286	1
Mozambique	Service Station	0.004543055556	2
Mozambique	Grocery Store	0.004218316327	3
Mozambique	Wholesale	0.001482083333	4
Namibia	Hotel/Lodge/Accomodation	0.236017777778	1
Namibia	Pub/Restaurant	0.126894629630	2
Namibia	Petrol Station	0.121662132765	3
Namibia	Grocery Store	0.101058491688	4
Namibia	Liquor Store	0.097950931954	5
Nigeria	Grocery Store	0.002218658537	1
South Africa	Sweet Shop	0.184861818182	1
South Africa	Hotel/Lodge/Accomodation	0.169117152917	2
South Africa	Tarven/Bar	0.153727500000	3
South Africa	Bottle Store	0.151201594203	4
South Africa	Pub/Restaurant	0.124997867926	5
South Africa	Wholesale	0.118420195882	6
South Africa	Take Away	0.118240000000	7
South Africa	Tobacco Shop	0.106306000000	8
South Africa	Petrol Station	0.094685380644	9
South Africa	Grocery Store	0.090688719516	10
South Africa	Liquor Store	0.083416002700	11

Fig. 7. Partial output of query 4

E. Which countries consume more tobacco per capita?

```
SELECT
    sales.country,
    SUM(sales.quantity) / population.total_pop AS
    sales_per_capita
FROM ( SELECT year_id,
            country_id,
            country,
            SUM(quantity) AS quantity
        FROM
            sales s
        INNER JOIN
            dim_time t USING (day_id)
  INNER JOIN
            dim_location l USING (suburb_id)
        WHERE
            t.year = 2019
        GROUP BY
            t.year_id,
            l.country_id,
            l.country) AS sales
INNER JOIN ( SELECT
                p.country_id,
                p.year_id,
                total_pop
             FROM
                population p
            INNER JOIN
                dim_year y USING (year_id)
            WHERE
                y.year = 2019) AS population
ON
    sales.country_id = population.country_id AND
    sales.year_id = population.year_id
GROUP BY
    sales.country_id,
    sales.country,
    population.total_pop
ORDER BY
    2 DESC;
```

This query is designed to calculate the per capita sales of cigarettes for the year 2019. The choice of 2019 was based on the availability of data across multiple countries in that particular year. The query involves merging two tables, which were not directly connected through a shared dimension. Consequently, it was necessary to perform two separate queries and then join them based on the year_id to establish a connection. The first subquery retrieves the total quantity of cigarettes sold per country in 2019. It aggregates the sales data by grouping it by year_id and country_id. The second subquery retrieves the total population for each country in 2022. It associates each country with its respective population data using the country_id and year_id. The main query then joins these two subqueries based on the shared country_id and year_id.

country	sales_per_capita
Namibia	0.06819281840765994
Botswana	0.05129111390077697
Lesotho	0.029974812441198328
South Africa	0.0055176794554311626
Zimbabwe	0.00307920592958153
Malawi	0.0019916430177719306
Eswatini	0.00039072753124324026
Mozambique	0.0002745199491705545
Zambia	0.00027387754953258286
Ethiopia	0.00020963788534083516
Tanzania	0.00017771073465868 24

Fig. 8. Partial output of query 6

F. What is the biggest brand in each country?

```
SELECT
    l.Country,
    p.Brand,
    SUM(Price) AS 'Sales Value',
    (SUM(Price) * 100 / total.total_sales) AS '
    Market Share',
    RANK() OVER (PARTITION BY
            Country
        ORDER BY
            (SUM(Price)) DESC
    ) AS 'Market Share Rank'
FROM
    dw_cigarettes.sales s
INNER JOIN
    dw_cigarettes.dim_location l
ON
    s.Suburb_ID = l.Suburb_ID
INNER JOIN
    dw_cigarettes.dim_product p
ON
    s.Product_ID = p.Product_ID
INNER JOIN (SELECT
        Country,
        SUM(Price) AS total_sales
    FROM
        dw_cigarettes.sales s
    INNER JOIN
        dw_cigarettes.dim_location l
    ON
        s.Suburb_ID = l.Suburb_ID
    GROUP BY
        Country) AS total
ON
    l.Country = total.Country
GROUP BY
    l.Country,
    p.Brand,
    total.total_sales
ORDER BY
    Country,
    SUM(Price) DESC;
```

To answer this question we needed to develop a more involved query. On top of joining the relevant tables we have in the dataset, we also had to create an auxiliary table that would allow us to extract the value of sales in total per country. With this, we were able to create a column that gives us the market share of each brand in each country. Similarly to the query for question *D*, we also added a column that would explicitly state the ranking at which each brand would be in terms of market share for the specific country in question by using a partition by country. As in previous cases, in Figure 8 we show a section of the output of the query and, just for visual purposes when inserting the partial output in this report, we added an additional *HAVING* condition in order to only show brands with market share over 5%.

Country	Brand	Sales Value	Market Share	Market Share Rank
Botswana	Peter Stuyvesant	19683.9252	46.75982296	1
Botswana	Dunhill	6176.0596	14.67143624	2
Botswana	Marlboro	3091.7336	7.34451662	3
Botswana	Craven A	2893.6734	6.87401799	4
Cameroon	Oris	10.1183	20.49138994	1
Cameroon	Time	9.1740	18.57901143	2
Cameroon	Esse	7.5553	15.30085078	3
Cameroon	Fine	5.3967	10.92929485	4
Cameroon	Rothmans	3.5976	7.28579153	5
Cameroon	D & J	3.3728	6.83053082	6
Cameroon	Manchester	2.5633	5.19114672	7
Chad	Fine	280.0567	44.55620662	1
Chad	Manchester	155.3565	24.71676740	2
Chad	Oris	47.2646	7.51966042	3
Chad	Dunhill	33.2793	5.29463986	4
Eswatini	Dunhill	720.2003	28.90878223	1
Eswatini	Peter Stuyvesant	552.6254	22.18233920	2
Eswatini	Marlboro	217.8955	8.74630788	3
Eswatini	Chesterfield	162.6821	6.53004644	4

Fig. 9. Partial output of query 5

V. Data Analysis

Upon successfully populating the necessary databases, we proceeded to import our dataset into an Excel file. This was accomplished through establishing a connection between Power Query and MySQL. Once the connection was secured, we utilized Data Analysis Expressions (DAX) within Excel to create relationships among the underlying datasets. With these dataset connections firmly in place, we were able to construct the subsequent reports:

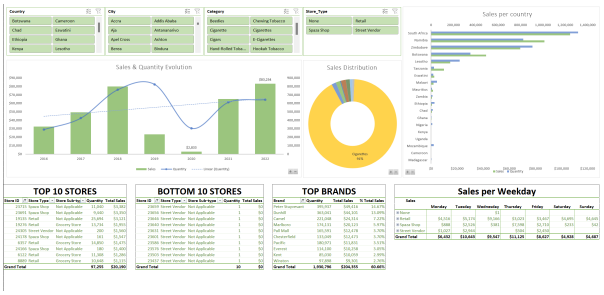

Fig. 10. Dashboard - Sales view

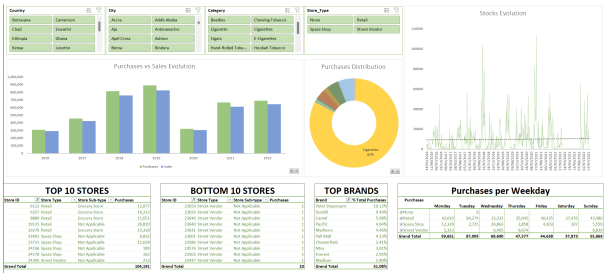

Fig. 11. Dashboard - Purchases & Stocks view

VI. Comparison With Operational Database

Data Warehouses and Standard Operational Databases serve distinct purposes in the topic of data management, each with its own set of advantages and shortcomings. Understanding these differences is crucial for each organization to make informed decisions regarding their data infrastructure. On this section, we go through the advantages and shortcomings of both data warehouses and operational databases.

A. Advantages of Data Warehouses

Data warehouses excel in facilitating strategic decision-making processes and this is usually the main purpose for their existence. By integrating data from multiple sources, potentially an operational database, and transforming it into a structured format, data warehouses provide a unified view of the organization's data. This enables decision-makers to analyze historical trends, identify patterns, and derive actionable insights, crucial for their planning. This quality comes from the fact that this type of structure is designed to allow simple and direct queries and analytical processing. Through procedures like indexing, partitioning, and aggregations, data warehouses optimize query performance even on extremely large datasets. This is particularly beneficial for generating reports, conducting analysis, and performing ad-hoc queries without impacting the operational databases' performance and structure. Data warehouses typically undergo through processes such as data cleansing and transformation, ensuring high data quality and consistency. By consolidating data from various sources and standardizing formats, data warehouses minimize inconsistencies and inaccuracies, providing users with reliable information.

B. Shortcomings of Data Warehouses

One of the primary shortcomings of data warehouses is data latency. Due to the ETL (Extract, Transform, Load) processes involved in populating data warehouses, there is often a delay between data generation and its availability for analysis. This latency can limit the timeliness of insights, especially in fast-paced environments where real-time data analysis might be needed. Also, implementing and maintaining a data warehouse infrastructure can be costly and complex. From hardware and software investments to ongoing maintenance and resource allocation, organizations incur significant expenses. Additionally, designing and managing the ETL processes require specialized skills and resources, adding to the complexity and overhead. Lastly, a data warehouse is not very flexible in the sense that once the data is loaded, there is no room for constant updates during the day-to-day operations of a business. For this, the best choice would be an operational database.

C. Advantages of Operational Databases

Operational databases are optimized for transactional processing, enabling real-time data insertion, updates, and retrieval. This capability is essential for supporting mission-critical applications that require immediate access to the most up-to-date information, such as online transaction processing

(OLTP) systems. With their focus on transactional operations, operational databases offer agility in handling dynamic and rapidly changing data requirements, making them well-suited for operational tasks and transactional workflows.

D. Shortcomings of Standard Operational Databases

Operational databases are not optimized for complex analytical queries and reporting. While they excel in transactional processing and efficiency (by avoiding redundancies and repeated information), their performance may degrade when subjected to intensive analytical workloads. They are usually difficult to query which limits their suitability for strategic decision-making and in-depth data analysis tasks.

In summary, both data warehouses and standard operational databases play vital roles in an organization's data ecosystem, each catering to specific use cases and requirements. While data warehouses are denormalized, analysis oriented, and offer centralized data management, they come with inherent challenges such as data latency and complexity. On the other hand, operational databases are highly normalized, excel in transactional processing and agility but may lack the analytical firepower needed for strategic decision-making. Therefore, organizations must carefully evaluate their objectives, data needs, and resource constraints to determine the optimal balance between data warehousing and operational database solutions. In what concerns our application, an operational database could be useful to store the daily transactional activity of a store, for instance. In this database, the system could record information on each product it sells, each client, supplier, and many other factors of the day-to-day activity in a highly normalized and efficient way. Specifically for the dataset we based ourselves for this project, and given the global and high-level scope it brings, it would not realistically make sense to use such an architecture for structuring the data. This dataset was essentially tailored for data analysis and querying for a number of metrics and trends of the cigarette market in many countries of Africa and this would not be easily and directly achieved using an operational database.

VII. Key takeaways, insights and proposals

During this work we aimed to create a data warehouse tailored for the cigarettes market in Africa, starting by structuring the design of the data warehouse, organized around the store, location, country, time, year, and product dimensions, and also facts like sales, purchases, stocks and population. This way of modeling proved effective in centralizing and organizing the dataset.

The ETL process had a crucial role in converting the raw dataset into a usable format for the data warehouse. The use of different data sources and techniques like data cleaning, transformation, and generation of additional features such as stocks and purchases enriched the dataset. All this process allows us to go into the next step - querying. Here, one of the major advantages of data warehousing becomes clear: making a query in this format becomes much simpler, enabling the analysis of the data in a quicker way. However, it is also necessary to be aware of the disadvantages. The data warehouse format is not the best one when we are talking about transactional processing and real-time data management, being the operational databases the best option for this case. Briefly, while data warehouses are better suited for analytical tasks and strategic decision-making, for operating real time programs, an operational database is usually the best option. In addition, we promote the visualization of the dataset for a better understanding of the data. For that purpose, two dashboards were created in Microsoft Excel, one for sales and another one purchases and stocks. Here is where relevant information can be seen, like the amount of sales, and the best and worst selling shops/brands.

For future work, this data warehouse can evolve with the addition of more external data sources and with advanced analytical techniques such as predictive modeling and machine learning to forecast market trends. Implementing a streamlined ETL process, triggered seamlessly by the population of the sales table, taking into example the stocks generation, would also be an improvement for our warehouse. By eliminating the need for checking the data availability and enabling immediate action upon sales data ingestion, we ensure timely and accurate generation of stock information. This approach would also reduce operational costs since the process would be running once, in less time, instead of checking X times.

References

[1] Kaggle dataset url - https://www.kaggle.com/datasets/waalbannyantudre/african-cigarette-prices

[2] The world bank url - https://data.worldbank.org/