Data Warehousing

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Kavya Nagpal (181269)
Shreya Chaudhary (181279)

Under the supervision of

Prof.(Dr.) Ravindara Bhatt
to

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

Certificate

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Data Warehousing"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2021 to December 2021 under the supervision of **Prof.(Dr.) Ravindara Bhatt,** Associate Professor, Department of Computer Science and Engineering.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Kavya Nagpal (181269)

Shreya Chaudhary (181279)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(College Supervisor Signature)

Supervisor Name: Ravindara Bhatt

Designation: Associate Professor

Department name: Computer Science and Information Technology

Dated: 25th May, 2022

(Company Supervisor Signature)

Supervisor Name: Alok Choudhary

Designation: Manager

Department name: BT, ZS Associates

Dated: 25th May, 2022

ACKNOWLEDGEMENT

We are really grateful and wish our profound indebtedness to our project Supervisor Prof.(Dr.) Ravindara Bhatt, Associate Professor, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Profound knowledge and proficiency of our supervisor in his field has helped us to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would also generously welcome each one of those individuals who have helped us in making this project a win. In this unique situation, we might want to thank the various staff individuals, both educating and non-instructing, who have developed their convenient help and facilitated our undertaking.

Finally, We must acknowledge with due respect the constant support and patience of our parents.

Kavya Nagpal(181269)
Shreya Chaudhary(181279)

Table of Contents

TABLE OF FIGURES

TABLE OF GRAPHS

ABSTRACT

Data warehousing is a data management system that is designed to enable and support business intelligence (BI) activities like analytics. Data repositories generally contain large historical data and are intended only for the purpose of questioning and analysis. Considered a core component of Business Intelligence(BI), a Data Warehouse(DW) is used for data analysis as well as reporting. DWs are centralized collections of integrated data from one or more different sources.

Data View, which is a graphic display of information and data, provides an accessible way to visualize and understand trending, external, and patterns in data by using visual features such as charts, graphs, and maps, data display tools. Data visualization makes data easier for the human brain to comprehend and retrieve information by translating information into a visual context, such as a map or graph. The main goal of data viewing is to make it easier to identify patterns, trends and external products in large data sets.

Chapter-1

INTRODUCTION

1.1 Introduction

- Pharma ABC was established in 1964 and is headquartered at New York
- Pharma ABC is well known for key innovations and always introduces drugs in new markets
- It has launched 25 drugs so far of which 7 are blockbusters
- ZS team worked closely with these internal teams to identify their reporting needs

As a next step, the client decided to seek our help in using the existing system to generate four reports and integrate them into the dashboard.

The pharmaceutical industry acquires, develops, manufactures, and markets drugs or drugs to be used as drugs to be given to patients (or to administer them), for the purpose of self-treatment, vaccination, or reduction of symptoms.

The United States continued to be the world's largest drug market in 2019, with revenue close to billion. Database stores data aggregated from singular or multiple data sources which are remote for query and analysis. The information integrated into the database is stored as visual aids. Viewing is a visual relationship defined using real relationships stored on a website. Physical observation is the result of a study of the related algebraic condition that describes the visual correlation. By using these 2 body-generated observations, user requirements and questions can be answered quickly and information can be obtained either directly or calculated using these modified physical views. A problem which is also known as the viewing problem is a way to save the modified view to be updated with information about actual updates in relationships to remote data sources. There are many algorithms that are developed to solve the issue with viewing traditional data systems. In applications, questionnaires outlined by ideas and real relationships are stored on the same website. Web systems understand visual definitions and know what data is needed to distribute updates to ideas whereas in a DW, queries describing views and actual relationships can be stored on a separate location such as a website located on multiple sites. Sources may notify the data warehouse in the event of a review but may not be able to determine which data is

required to update comments in the database. They can therefore only send real data updates or all updated relationships to the database. After getting insights like these, the data repository may find that it requires additional source data to review the view which is when it draws queries from other sources to request some other related additional source data. Some sources may review its data again before reviewing query requests in the data store. They will therefore send additional unrelated and incorrect data to the database, which will use it to calculate ideas. This condition is called a distributed maintenance anomaly. Solving the archive view problem is therefore much more complex.

1.2 Problem Statement

The data provided has to be analyzed, cleaned and visualized using a data visualization tool tableau.

The visualization requirements were as follows:

- Track market share of products

  Solution centric approach to find out the market share of the product ,leveraged technology and visualization of the trend.

- Track sales of different products

  Find out the TRx sales of the products

- Track top 5 territories in the market

  Chart out the top 5 territories in the market

- Track top 10 HCPs in the market

  Find out the top 10 HCPs based on the market from the reporting table

1.3 Objectives

1. Gathering data files containing raw data.

2. Loading the files into the database Redshift for data cleaning and modeling.

3. Apply DQM checks for data cleaning.

4. Make Staging layer of data and make facts , dimensions and reporting layers

5. Visualize data for the given problem statements using Tableau

Procedure

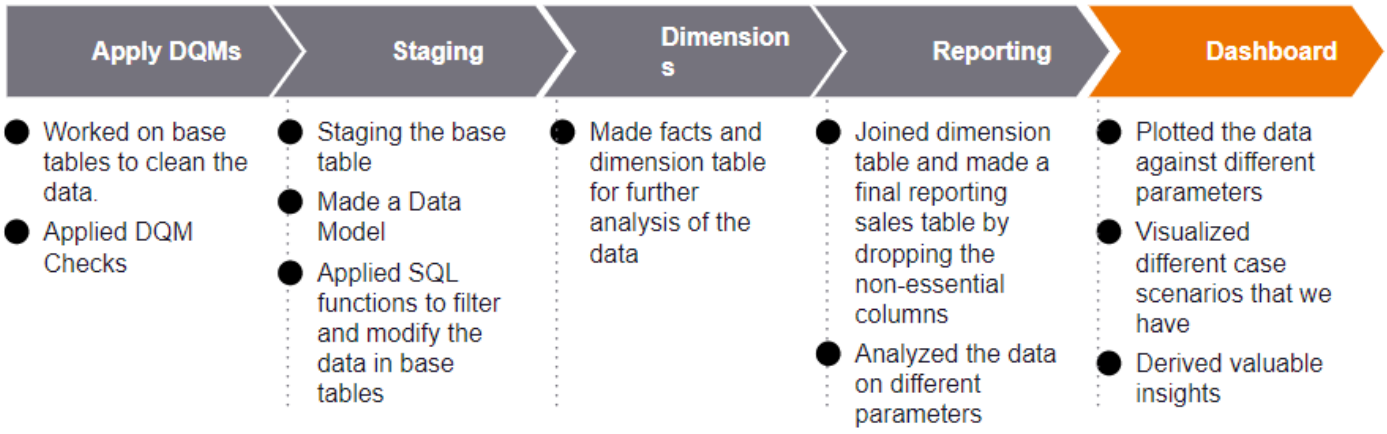


Figure 1

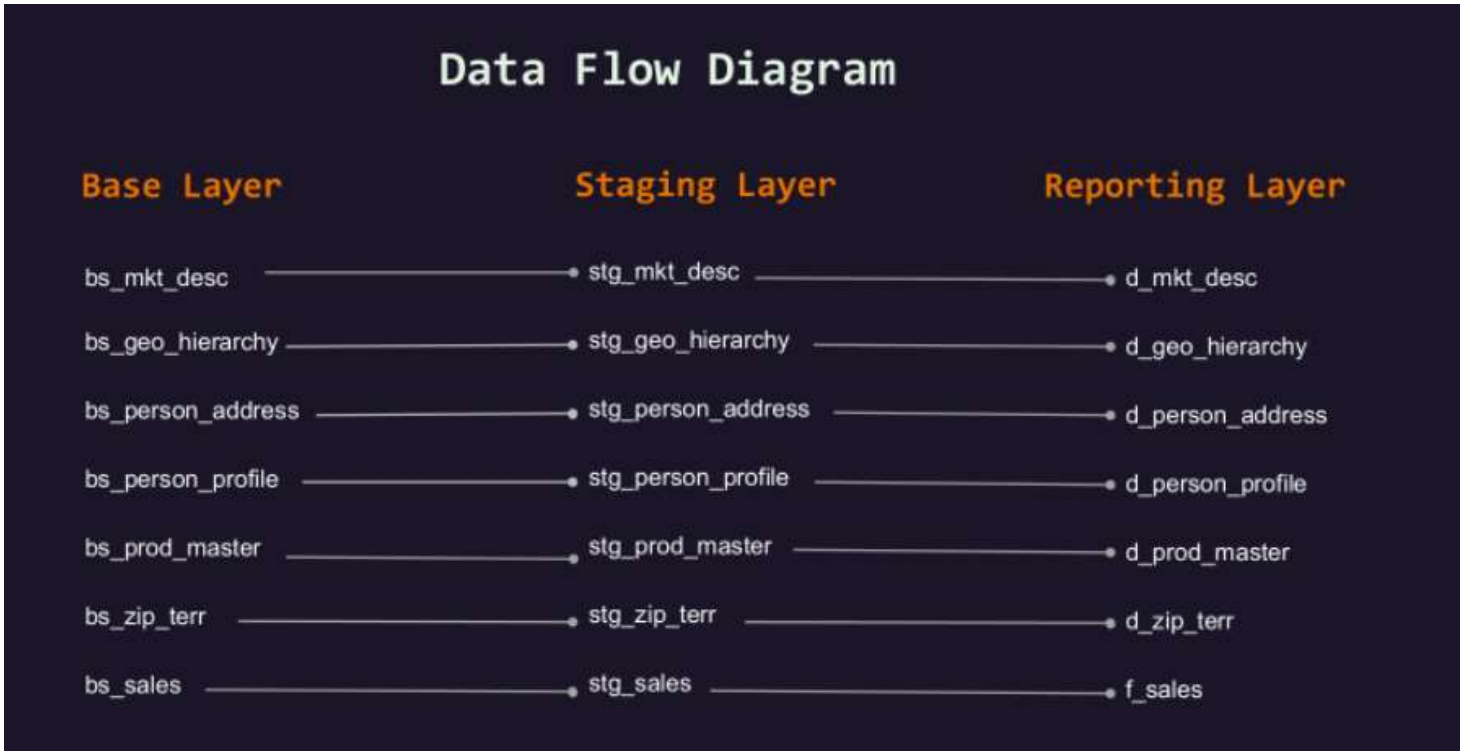


Figure 2

1.4 Methodology

The entire process was divided into 4 phases. Requirement gathering and data collection, data cleaning, data modeling and warehousing and data visualization.

1.4.1 Datasets and inputs

Data Source Name	Description
Geo_Hierarchy.txt	This file defines the relation between territories and their parent (mapping a territory to a district which is then mapped to a region)
Mkt_Desc.txt	This file maps the market code to the market name.
Person_Address.txt	This file provides the address and zip code of the prescriber.
Person_Profile.txt	This file provides information about the specialty of a prescriber.
Prod_Master.txt	This file gives information about the client's as well as their competitor's products in the market.
Prod_Team.txt	This file gives information about which team is promoting which product.
Sales.txt	The file contains detailed information about the sales of the products segregated in terms of TRx, NRx divided in terms of the volume, units and dollar sales.
Zip_Terr.txt	This file contains the mapping between the different territories and the teams.

1.4.2 Data Preprocessing

- Data layer[Raw] -The main aim of this layer is to ingest raw data into the DW as quickly and as efficiently as possible. To do so, data should remain in its native format. We don't allow any transformations at this stage. With Raw, we can get back to a point in time, since the archive is maintained. No overriding is allowed, which means handling duplicates and different versions of the same data. Despite allowing the above, Raw still needs to be organized into folders. From our experience we advise customers to start with generic division: subject area/data source/object/year/month/day of ingestion/raw data. It is important to mention that end users shouldn't be granted access to this layer. The data here is not ready to be used, it requires a lot of knowledge in terms of appropriate and relevant consumption. Raw is quite similar to the well-known DWH staging.
- Data layer[Standardized] – It is not necessary in most DW systems. If we anticipate that our Data Lake Architecture will grow fast, this is the right direction. The main objective of this layer is to improve performance in data transfer from Raw to Curated. Both daily transformations and on-demand loads are included. While in Raw, data is stored in its native format, in Standardized we choose the format that fits best for cleansing. The structure is the same as in the previous layer but it may be partitioned to lower grain if needed.
- Data layer[Cleansed] - Here, data is transformed into data sets that are useful and consumable is to be stored in files or tables. The purpose of the data, as well as its structure at this stage is already known. You should expect cleansing and transformations before this layer. Also, denormalization and consolidation of different objects is common. Due to all of the above, this is the most complex part of the whole Data Lake solution. In regards to organizing your data, the structure is quite simple and straightforward. For example: Purpose/Type/Files. Usually, end users are granted access only to this layer.
- Data layer[Application] - Here, data is sourced from Cleansed data layer and enforced with any needed business logic that the user might have specified. These might be surrogate keys shared among the application, row level security or anything else that is specific to the application consuming this layer. If any of your applications use machine learning models that are calculated on your Data Lake, you will also get them from here. The structure of the data will remain the same, as in Cleansed.
- Data layer[Sandbox] It is a somewhat optional layer, specifically meant for advanced analysts and data scientists related work. Here they can carry out their experiments when looking for patterns or correlations. Whenever you have an idea to enrich your data with any source from the Internet, Sandbox is the proper place for this.

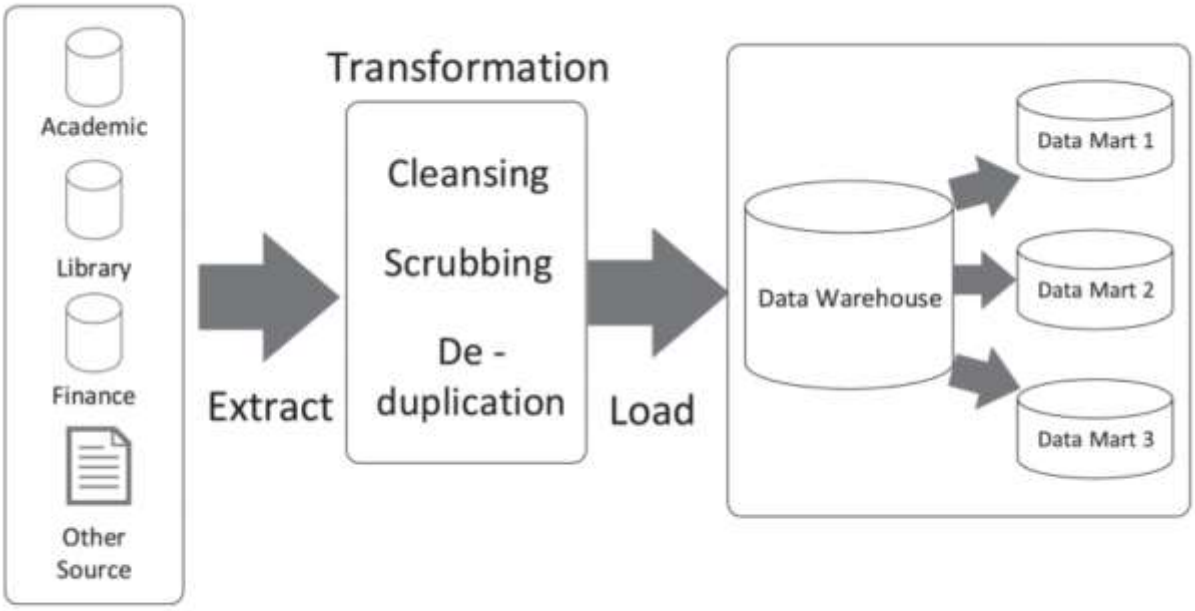


Figure 3

1.5 Organization

- Security –Even though data lakes are not disclosed to a very wide audience, it is still pretty much important to consider this feature, especially during the initial phase and design. It is not the same as related information, which is armed with security measures. It is always better to be careful about this without ignoring future requirements.
- Governance – Managing, monitoring, and listing of data is necessary from time to time to measure the performance and maintenance of our Data Lake.
- Metadata – Metadata is the data about data, and is information about every scheme, every reload interval, any additional description of the purpose of the data, and description of how it is intended to be utilized.
- Stewardship – depending on the size that is required by us, either a separate group is made or this responsibility is transferred to the owners (users), perhaps by using other metadata solutions.
- Master Data – an important part of providing data that is completely ready to be used .

- Archive – if there is another DWH related solution, the user might face some issues related to working with local storage. Data Lake is often used to store specific archive data from DWH.
- Offload – if there are some other DWH related solutions, the user may want to use this site to deploy ETL processes / resources that can consume a lot of time, in your Data Lake, to make it cheaper and faster.
- Orchestration + ELT processes – as we get data from the previous Layer we use the Filter for our last two layers i.e. the Sandbox and Application Layer. It is very much possible, that we may have to apply the changes and therefore we can either choose an orchestration tool that can do so, or some additional added on resources to use it.

CHAPTER-2

LITERATURE SURVEY

List of Research Papers and Journals

Title	A Study on Real-Time Detection Method of Lane and Vehicle for Lane Change Assistant System Using Vision System on Highway
Authors	VanQuangNguyen ,HeungsukKim, SeoChangJun, KwangsuckBoo
Year of Publication	February 2016
Web Link	https://doi.org/10.1016/j.jestch.2018.06.006

Title	Lane detection for driver assistance and intelligent vehicle applications. 2007 international symposium on communications and information technologies. 1291-1296
Authors	Craig D'Cruz, Ju Jia Zou
Year of Publication	March 2016
Link	https://ieeexplore.ieee.org/document/4392216

Title	CNN based lane detection with instance segmentation in edge-cloud computing
Authors	Wei Wang, Hui Lin ,Junshu Wang
Year of Publication	June 2020
Link	https://doi.org/10.1186/s13677-021-00267-1

Chapter-3

SYSTEM DEVELOPMENT

3.1 Model Development

The first layer that is the L0 layer/ Landing Layer:

This is the layer in which the data is raw. It is the data which is brought in directly from the source or from the data lake. Nothing is done to the data in this layer. It is maintained in its original form.

The second layer that is the L1/ Staging layer

This is the layer in which we perform the pre dqm and dqm checks.

Importing the required libraries

```
In [78]: import numpy as np
         import pandas as pd
         pd.set_option('display.max_columns', None)
         pd.set_option('display.max_rows', None)
         import matplotlib.pyplot as plt
         %matplotlib inline
```

Figure 4

Loading all the dataframes

```
In [23]: mkt_desc=pd.read_csv('bs_mkt_desc.txt',delimiter='\t')
         geo_hierarchy=pd.read_csv('bs_geo_hierarchy.txt',delimiter='\t')
         hierarchy=pd.read_csv('bs_hierarchy.txt',delimiter='\t')
         person_profile=pd.read_csv('bs_person_profile.txt',delimiter='\t')
         person_address_udp=pd.read_csv('bs_person_address_udp.txt',delimiter='\t')
         prod_master=pd.read_csv('bs_prod_master.txt',delimiter='\t')
         prod_team=pd.read_csv('bs_prod_team.txt',delimiter='\t')
         sales=pd.read_csv('bs_sales.txt',delimiter='\t')
         zip_terr=pd.read_csv('bs_zip_terr.txt',delimiter='\t')
```

Figure 5

Mkt_desc table has no nulls or duplicates and therefore it is pushed to the staging layer directly.

1. Mkt_desc

```
In [4]: # no nulls/duplicates
        #converting to staging table

        mkt_desc
```

Out[4]:

	MKT_CD	MKT_NM
0	1	NEUROSCIENCE
1	2	VIROLOGY

```
In [5]: mkt_desc.to_csv('stg_mkt_desc.csv')
```

Figure 6

The geo hierarchy table had some empty rows for column Parent_ID which we filled as unassigned for data standardization.

```
In [8]: geo_hierarchy['PARENT_ID'] = geo_hierarchy['PARENT_ID'].fillna('unassigned')

In [9]: geo_hierarchy.isna().any() #checking after filling

Out[9]: TERR_CODE          False
        TERR_NM            False
        TERRITORY_LEVEL    False
        PARENT_ID          False
        dtype: bool

In [10]: geo_hierarchy[geo_hierarchy['TERRITORY_LEVEL']=='NATION']
```

Out[10]:

	TERR_CODE	TERR_NM	TERRITORY_LEVEL	PARENT_ID
244	C00000	Nation	NATION	unassigned

```
In [11]: geo_hierarchy.to_csv('stg_geo_hierarchy.csv')
```

Figure 7

The hierarchy table had some duplicates which were dropped,

```
In [14]: hierarchy = hierarchy.drop_duplicates(["TERR_ID","TERR_NM","DIST_ID"],keep="first

In [15]: hierarchy[hierarchy.duplicated(["TERR_ID","TERR_NM","DIST_ID"],keep=False)]  # du
```

Out[15]:

TERR_ID	TERR_NM	DIST_ID	DIST_NM	REGN_ID	REGN_NM

```
In [16]: hierarchy.shape

Out[16]: (238, 6)

In [17]: hierarchy.to_csv('stg_hierarchy.csv')
```

Figure 8

We dropped the gender and email columns in person_profile as they did not have any entry.

```
In [24]: person_profile = person_profile.drop(['GENDER' ,'EMAIL'] , axis =1)

In [25]: person_profile.columns

Out[25]: Index(['PERSON_CODE', 'FIRST_NAME', 'LAST_NAME', 'PREFFERED_NAME',
                'PERSON_SPECIALTY'],
               dtype='object')

In [26]: person_profile.isna().any()

Out[26]: PERSON_CODE         False
         FIRST_NAME          False
         LAST_NAME           False
         PREFFERED_NAME      False
         PERSON_SPECIALTY    False
         dtype: bool

In [27]: person_profile.duplicated().sum()

Out[27]: 0

In [28]: person_profile.to_csv('stg_person_profile.csv')
```

Figure 9

```
In [4]: person_address_udp=person_address_udp.drop(['ADDRESS_2'],axis=1)

In [5]: person_address_udp.columns

Out[5]: Index(['PERSON_CODE', 'ADDRESS_1', 'CITY', 'STATE', 'ZIP_CODE', 'PRI_ADDR'], dt
        ype='object')

In [6]: person_address_udp['STATE']=person_address_udp['STATE'].str.upper()
        person_address_udp['CITY']=person_address_udp['CITY'].str.upper()

In [7]: person_address_udp[person_address_udp['ZIP_CODE'] == 674]
```

Out[7]:

	PERSON_CODE	ADDRESS_1	CITY	STATE	ZIP_CODE	PRI_ADDR
321	327	1 & 102 MANATI MEDICAL	MANATI	PR	674	Y
1190	1167	1 CALLE MARGINAL	MANATI	PR	674	Y
1605	1583	1 CALLE JOSE D CANDELA	MANATI	PR	674	Y
1606	1584	1 CALLE JOSE D CANDELAS	MANATI	PR	674	Y
1607	1585	1 CALLE JOSE D CANDELAS	MANATI	PR	674	Y
1608	1586	1 CALLE JOSE D CANDELAS MANATI MEDICAL PLAZA S...	MANATI	PR	674	Y
2005	1980	1 Calle Jose D Candelas	MANATI	NaN	674	Y
13225	13235	1 TORRE MEDICA DR PEDR	MANATI	PR	674	Y
13226	13236	1 TORRE MEDICA DR PEDRO BLANCO	MANATI	PR	674	Y

Figure 10

In [11]:
```
#record where state was null is now filled with the correct state
person_address_udp[person_address_udp['CITY']== 'MANATI']
```

Out[11]:

	PERSON_CODE	ADDRESS_1	CITY	STATE	ZIP_CODE	PRI_ADDR
321	327	1 & 102 MANATI MEDICAL	MANATI	PR	674	Y
1190	1167	1 CALLE MARGINAL	MANATI	PR	674	Y
1605	1583	1 CALLE JOSE D CANDELA	MANATI	PR	674	Y
1606	1584	1 CALLE JOSE D CANDELAS	MANATI	PR	674	Y
1607	1585	1 CALLE JOSE D CANDELAS	MANATI	PR	674	Y
1608	1586	1 CALLE JOSE D CANDELAS MANATI MEDICAL PLAZA S...	MANATI	PR	674	Y
2005	1980	1 Calle Jose D Candelas	MANATI	PR	674	Y
2585	2557	1 CALLE MARGINAL	MANATI	PR	957	Y
13225	13235	1 TORRE MEDICA DR PEDR	MANATI	PR	674	Y
13226	13236	1 TORRE MEDICA DR PEDRO BLANCO	MANATI	PR	674	Y
44937	1030558	1 CALLE MARGINAL EXT	MANATI	PR	674	Y

In [12]:
```
person_address_udp.to_csv('stg_person_address_udp.csv')
```

Figure 11

```
In [42]: prod_master['PROD_COMP'] = prod_master['PROD_COMP'].fillna('N')
         prod_master['PROD_GEN'] = prod_master['PROD_GEN'].fillna('N')
         prod_master['PROD_CLIENT'] = prod_master['PROD_CLIENT'].fillna('N')

In [43]: prod_master.isna().any()

Out[43]: MARKET              False
         IMS_PRD_ID          False
         PRODUCT_NAME        False
         IMS_STRENGTH_LEVEL  False
         PRODUCT_CD          False
         NDC                 False
         PROD_GEN            False
         PROD_CLIENT         False
         PROD_COMP           False
         dtype: bool

In [44]: prod_master.to_csv('stg_prod_master.csv')
```

Figure 12

7. Product team

```
In [45]: # no nulls/ duplicates
         prod_team.sample(3)
```

Out[45]:

	PROD_CD	TEAM
8	1007	S
5	1001	S
1	1004	C

```
In [46]: prod_team.to_csv('stg_prod_team.csv')
```

Figure 13

8. Sales

```
In [47]: sales.columns
```

```
Out[47]: Index(['PERSON', 'PRODUCT', 'MONTH_1_NRX', 'MONTH_2_NRX', 'MONTH_3_NRX',
       'MONTH_4_NRX', 'MONTH_5_NRX', 'MONTH_6_NRX', 'MONTH_7_NRX',
       'MONTH_8_NRX', 'MONTH_9_NRX', 'MONTH_10_NRX', 'MONTH_11_NRX',
       'MONTH_12_NRX', 'MONTH_1_NRX_UNIT', 'MONTH_2_NRX_UNIT',
       'MONTH_3_NRX_UNIT', 'MONTH_4_NRX_UNIT', 'MONTH_5_NRX_UNIT',
       'MONTH_6_NRX_UNIT', 'MONTH_7_NRX_UNIT', 'MONTH_8_NRX_UNIT',
       'MONTH_9_NRX_UNIT', 'MONTH_10_NRX_UNIT', 'MONTH_11_NRX_UNIT',
       'MONTH_12_NRX_UNIT', 'MONTH_1_NRX_DOLLAR', 'MONTH_2_NRX_DOLLAR',
       'MONTH_3_NRX_DOLLAR', 'MONTH_4_NRX_DOLLAR', 'MONTH_5_NRX_DOLLAR',
       'MONTH_6_NRX_DOLLAR', 'MONTH_7_NRX_DOLLAR', 'MONTH_8_NRX_DOLLAR',
       'MONTH_9_NRX_DOLLAR', 'MONTH_10_NRX_DOLLAR', 'MONTH_11_NRX_DOLLAR',
       'MONTH_12_NRX_DOLLAR', 'MONTH_1_TRX', 'MONTH_2_TRX', 'MONTH_3_TRX',
       'MONTH_4_TRX', 'MONTH_5_TRX', 'MONTH_6_TRX', 'MONTH_7_TRX',
       'MONTH_8_TRX', 'MONTH_9_TRX', 'MONTH_10_TRX', 'MONTH_11_TRX',
       'MONTH_12_TRX', 'MONTH_1_TRX_UNIT', 'MONTH_2_TRX_UNIT',
       'MONTH_3_TRX_UNIT', 'MONTH_4_TRX_UNIT', 'MONTH_5_TRX_UNIT',
       'MONTH_6_TRX_UNIT', 'MONTH_7_TRX_UNIT', 'MONTH_8_TRX_UNIT',
       'MONTH_9_TRX_UNIT', 'MONTH_10_TRX_UNIT', 'MONTH_11_TRX_UNIT',
       'MONTH_12_TRX_UNIT', 'MONTH_1_TRX_DOLLAR', 'MONTH_2_TRX_DOLLAR',
```

```
In [48]: col = sales.columns[2:]
```

```
In [49]: col = list(col)
```

Figure 14

The third Layer is the L2 layer where we modify our data according to the user requirements

Our user requirements are as follows:

- Track market share of products

  Solution centric approach to find out the market share of the product ,leveraged technology and visualization of the trend.

- Track sales of different products

  Find out the TRx sales of the products

- Track top 5 territories in the market

  Chart out the top 5 territories in the market

- Track top 10 HCPs in the market

  Find out the top 10 HCPs based on the market from the reporting table

```
In [3]: sales = pd.read_csv('stg_sales.csv',index_col=[0])
        sales.head(3)
```

Out[3]:

	PERSON	PRODUCT	TRX	MONTH	TRX_UNIT	TRX_DOLLAR	NRX	NRX_UNIT	NRX_DOLLAR
0	11923	1010101030	172	1	3607	17200	38	779.182	3800
1	11931	1010101031	475	1	9958	47500	123	2563.880	12300
2	11935	1010101032	183	1	3840	18300	48	996.073	4800

```
In [7]: # sales=sales.drop(['NRX','NRX_UNIT','NRX_DOLLAR'],axis=1)
```

```
In [4]: sales.rename(columns={'PERSON':'PERSON_CD','PRODUCT':'IMS_PRD_ID'},inplace=True)
```

```
In [5]: sales.head(3)
```

Out[5]:

	PERSON_CD	IMS_PRD_ID	TRX	MONTH	TRX_UNIT	TRX_DOLLAR	NRX	NRX_UNIT	NRX_DOL
0	11923	1010101030	172	1	3607	17200	38	779.182	
1	11931	1010101031	475	1	9958	47500	123	2563.880	1
2	11935	1010101032	183	1	3840	18300	48	996.073	

Figure 15

Next we have the reporting table which will be used for data visualization.

```
 1   ADDRESS              517416 non-null  object
 2   CITY                 517416 non-null  object
 3   STATE                517416 non-null  object
 4   ZIP_CD               517416 non-null  float64
 5   PRI_ADDR             517416 non-null  object
 6   FIRST_NAME           517416 non-null  object
 7   LAST_NAME            517416 non-null  object
 8   PREFFERED_NAME       517416 non-null  object
 9   PERSON_SPECIALTY     517416 non-null  object
 10  IMS_PRD_ID           518400 non-null  int64
 11  TRX                  518400 non-null  int64
 12  MONTH                518400 non-null  int64
 13  TRX_UNIT             518400 non-null  int64
 14  TRX_DOLLAR           518400 non-null  int64
 15  NRX                  518400 non-null  int64
 16  NRX_UNIT             518400 non-null  float64
 17  NRX_DOLLAR           518400 non-null  int64
 18  PROD_CD              518400 non-null  int64
 19  TEAM                 518400 non-null  object
 20  MKT_CD               518400 non-null  int64
 21  PROD_NM              518400 non-null  object
 22  IMS_STRENGTH_LEVEL   518400 non-null  object
 23  NDC                  518400 non-null  object
 24  PROD_GEN             518400 non-null  object
 25  PROD_CLIENT          518400 non-null  object
 26  PROD_COMP            518400 non-null  object
 27  TERR_CD              517296 non-null  object
 28  TERR_NM              517296 non-null  object
 29  DIST_ID              517296 non-null  object
 30  DIST_NM              517296 non-null  object
 31  REGN_ID              517296 non-null  object
 32  REGN_NM              517296 non-null  object
dtypes: float64(2), int64(10), object(21)
memory usage: 134.5+ MB
```

```
In [62]: df3.to_csv('rpt_reporting_table.csv')
```

```
In [63]: fact_table = df3[['PERSON_CD', 'IMS_PRD_ID', 'PROD_CD','TERR_CD','ZIP_CD','MKT_CD
                'TRX_UNIT', 'TRX_DOLLAR', 'MONTH']]
```

```
In [64]: fact_table.to_csv('f_sales.csv')
```

Figure 16

3.2 Tools and Techologies

3.2.1 Redshift and SQL

Amazon Redshift is a data storage product that is part of Amazon Web Services' main platform. It is built on technology from the big data-compliant data company ParAccel, to handle large data sets and migration. MySQL Workbench combines management, SQL development,architecture, website design, and storage in an integrated, single development MySQL database system. It is a visual DB designing tool.

3.2.2 Tableau

Tableau Software is a data visualization software company which is based in America. Business Intelligence is the main domain on which the company is based on. The founding year of the was 2003 and it has its headquarters based in Washington, Seattle. The Company was acquired by another Business intelligence Company Salesforce in the year 2019. Data Visualizations and interactive dashboards can be made efficiently with Tableau and helps businesses understand data and take key decisions on its basis .

3.2.3 PYTHON

Python is an interpreted, high level, general-purpose programming language. Python is very easy to read and understand hence it is used widely. It has libraries such as Numpy ,Pandas and Pyspark which are capable of handling Big Data.

3.3 Proposed approach

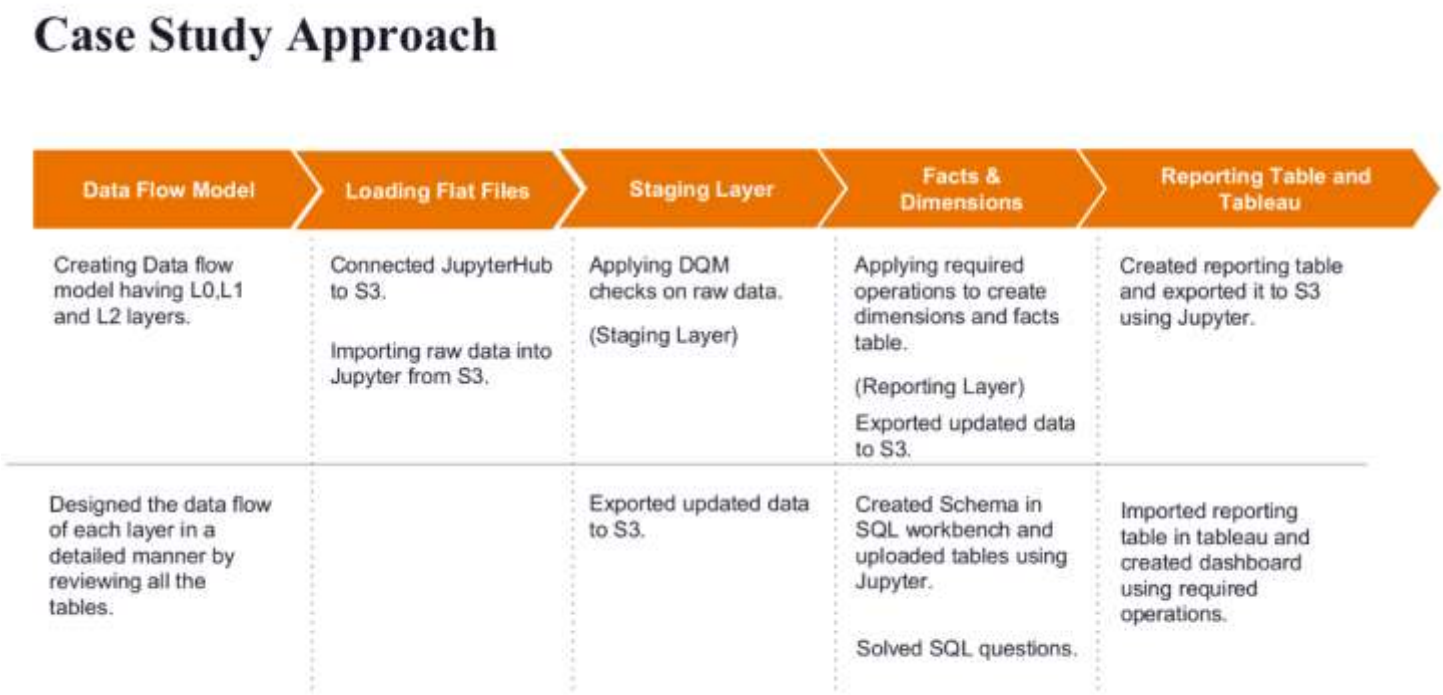


Figure 17

CHAPTER-4

PERFORMANCE ANALYSIS

Execution Times:

SQL server: CPU time = 424 ms

Elapsed time = 2300 ms.

It is therefore seen that we have a fairly accurate measure of how long it took for the query to execute and run and also how long it took to get parsed as well as compiled. The CPU time is a measure of how much CPU is used while the elapsed time tells how long it takes the query for overall execution.

All Sql queries are optimized.

Tableau performance Analysis

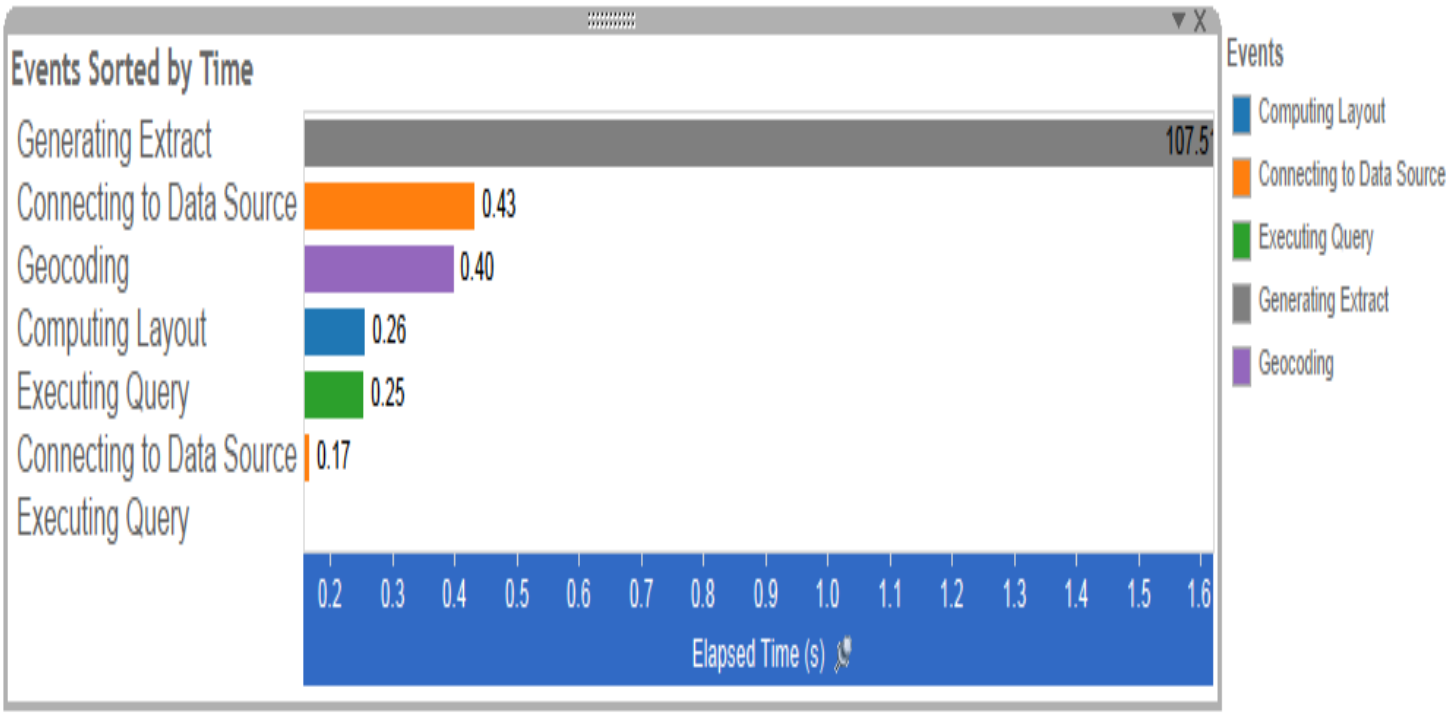


Figure 18

CHAPTER-5

RESULTS

SQl Analysis of Data

Find out the Top 10 Districts based on Sales from last year:

```
select dg.dist_nm, sum((fs.nrx_dollar + cast(fs.trx_dollar as float))) as sales
from d_geography_007 as dg
JOIN f_sales_007 as fs ON dg.terr_code = fs.terr_cd
group by dg.dist_nm
order by sales desc
limit 10;
```

dist_nm	sales
Mid Atlantic-Specialty	3966218000
Southeast-Specialty	3771354000
Unassigned	3745939200
West-Specialty	3445267200
Central-Specialty	3336288000
Northeast-Specialty	1672813600
Denver	1525666000
Los Angeles	1211762400
Mid South	1129211200
Houston	819064000

Figure 19

Find out the Sales trend of the products for last 4 quarters:

quarter_id	sales
Q1	4268566000
Q2	4271253000
Q3	4257483000

```
select quarter_id, sum((nrx_dollar + cast(trx_dollar as float))) as Sales
from rpt_sales_group7
group by quarter_id;
```

Figure 20

Calculate YTD and QTD sales for all the products in USA:

```
select sum(sales) as YTD_Sales
from rpt_sales_group7
where quarter_id = 'Q1';
```

ytd_sales
4268566000

```
select sum(sales) as QTD_Sales
from rpt_sales_group7
where month_id between 1 and 3;
```

qtd_sales
4268566000

Figure 21

Calculate the sales of a product by specialty type:

```
select  ndc, person_specialty, sum((nrx_dollar + cast(trx_dollar as float))) as p_sales
from rpt_sales_group7
group by  person_specialty, ndc
order by p_sales desc;
```

Figure 22

ndc	person_specialty	p_sales
BRUFEN	IM	187626000
ATRIPXEN	IM	186293600
KOMBIGYLZIN	IM	176298400
PERFORCIN	IM	172811400
CLOPIGEL	IM	162334800
VIMOXEX	IM	161972200
XOPENEX	FM	158325400
ATRIPXEN	FM	144428400
PROXEN	FM	141283200
KOMBIGYLZIN	FM	139444000
CLOPIGEL	FM	139321600
PERFORCIN	FM	138395600
FORADIL	FM	138043200
VIMOXEX	FM	137050200
DISPRINEX	FM	134611600
SEREVENT	FM	122099400
BRUFEN	FM	120527400
SARADONIL	FM	117816800
CLOPIGEL	OBG	109861600
KOMBIGYLZIN	PD	107185400
SARADONIL	OBG	104213600
FORADIL	PD	103759000
PROXEN	PD	103499800
XOPENEX	OBG	103471400
BRUFEN	PD	103470800
XOPENEX	PD	101569800
ATRIPXEN	OBG	100705800
SARADONIL	PD	100544600
PROXEN	OBG	100317200
CLOPIGEL	PD	98860000
FORADIL	OBG	98127800
VIMOXEX	OBG	98048000

Figure 23

Calculate the Market share of the products:

```
select ndc,
cast((sum(nrx_dollar + cast(trx_dollar as float))/(select sum(nrx_dollar + cast(trx_dollar as float)) from rpt_sales_007))*100 as varchar(10))+'%' as Market_Share
from rpt_sales_007
group by ndc;
```

ndc	market_share
VIMOXEX	8.36290453%
SARADONIL	8.31890780%
ATRIPXEN	8.40675360%
XOPENEX	8.33042408%
FORADIL	8.34282867%
SEREVENT	8.32158794%
KOMBIGYLZIN	8.31960625%
CLOPIGEL	8.34395487%
BRUFEN	8.28891178%
PROXEN	8.30486020%
PERFORCIN	8.32660251%
DISPRINEX	8.33265773%

Figure 24

Top 10 performing physicians in last 1 year based on sales:

```
select person_code, preferred_name, sum((nrx_dollar + cast(trx_dollar as float))) as Sales
from rpt_sales_group7
group by person_code, preferred_name
order by Sales desc
limit 10
```

person_code	preferred_name	sales
1256479	KATHLEEN WILLIAMS	2697800
1751946	MICHAEL A CORBIN	2673800
1619474	MARTIN B TANNER	2635200
1894863	RANDY B ROSS	2625800
4616889	ZOLLMAN KOMMOR	2618200
1344829	LEE YOSOWITZ	2603000
1907170	RICHARD FROEB	2602200
1948095	ROBERT CLENDENIN	2600000
1212783	JOSEPH LOEFFLER	2590400
2393087	VICTOR BRESSLER	2565400

Figure 25

Tableau Analysis of Data.

Sales Trends for Products

Chart insight:

- This chart depicts the total monthly sales trends of all the products.

Graph 1

Sales Trends for Products

Chart insight:

- This chart depicts the total sales of all the products.
- The top product is **'XOPENEX 350mg'.**

Graph 2

Market share of each Product

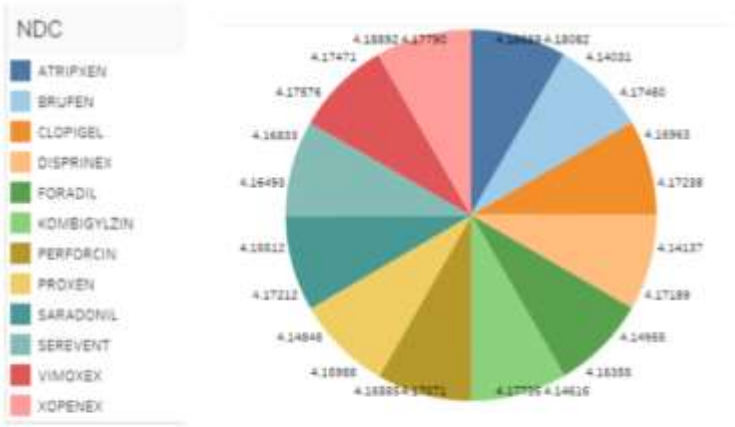


Chart insight:

- This chart depicts the market share of each product.
- The product with highest market share is **'ATRIPXEN 350mg'**.
- The product with lowest market share is **'PROXEN 350mg'.**

Graph 3

Market share of each Product in Neuroscience

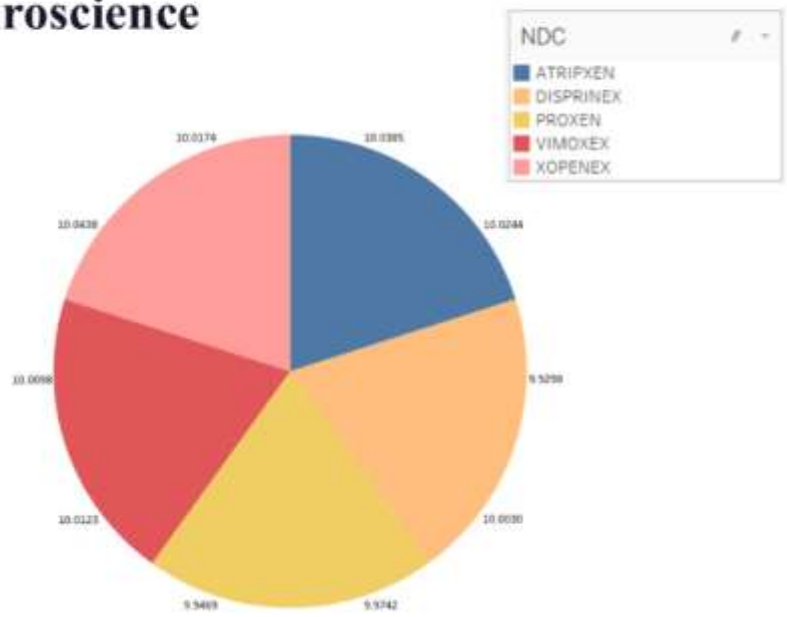


Chart insight:

- This chart depicts the market share of each product in the neuroscience market.
- The product with highest market share is **'XOPENEX 350mg'** with a market share of **10.04%**.
- The product with lowest market share is **'DISPRINEX 350mg'** with a market share of 9.93%.

Graph 4

Market share of each Product in Virology

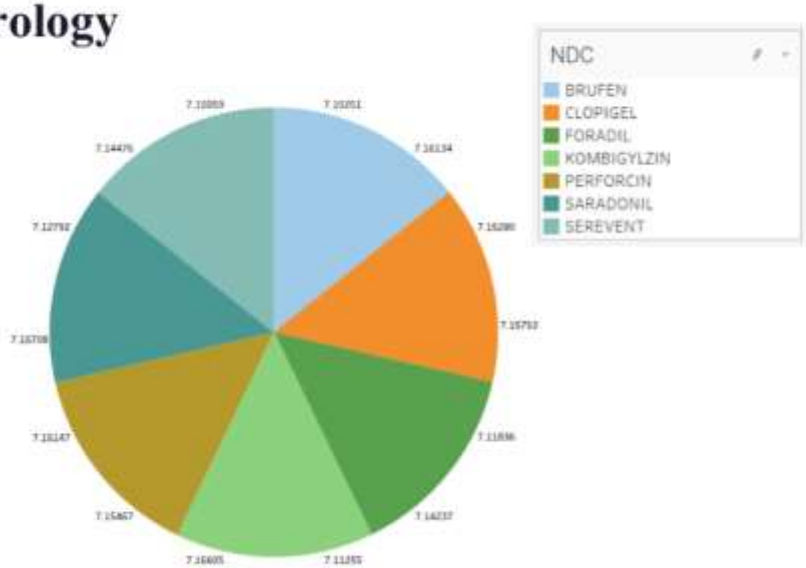


Chart insight:

- This chart depicts the market share of each product in the virology market.
- The product with highest market share is **'CLOPIGEL 500mg'** with 7.16% market share.
- The product with lowest market share is **'BRUFEN 350mg'** with 7.10% market share.

Graph 5

Top 5 Territories

Chart insight:

- This chart depicts the top 5 territories in the entire market.
- The market share is decided basis sales which is decided basis(TRX Dollar)
- The territory with highest sales is Los Angles South, CA

Graph 6

Top 5 Territories

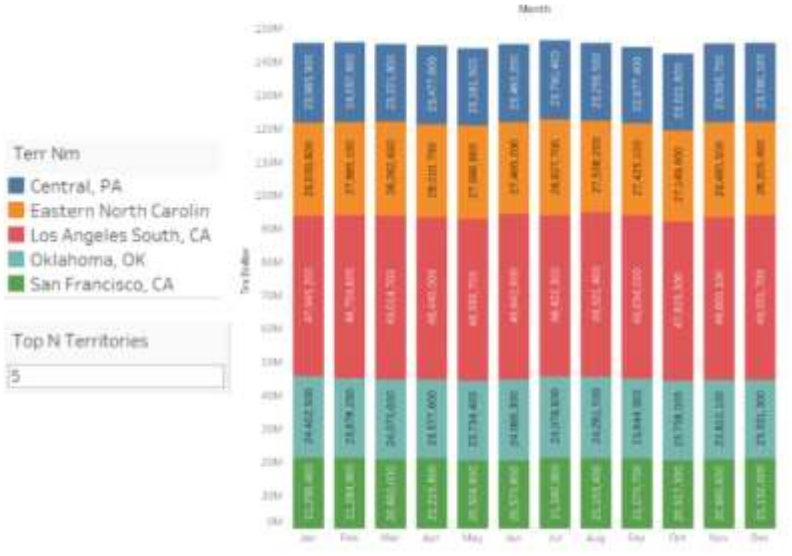


Chart insight:

- This chart depicts the monthly sales trends of top 5 territories in the entire market.

Graph 7

Top 5 Territories

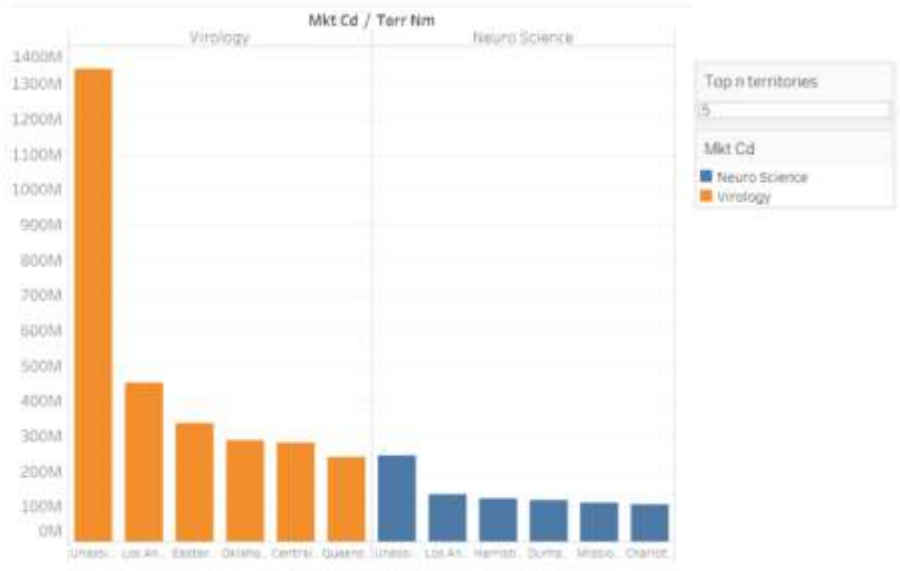
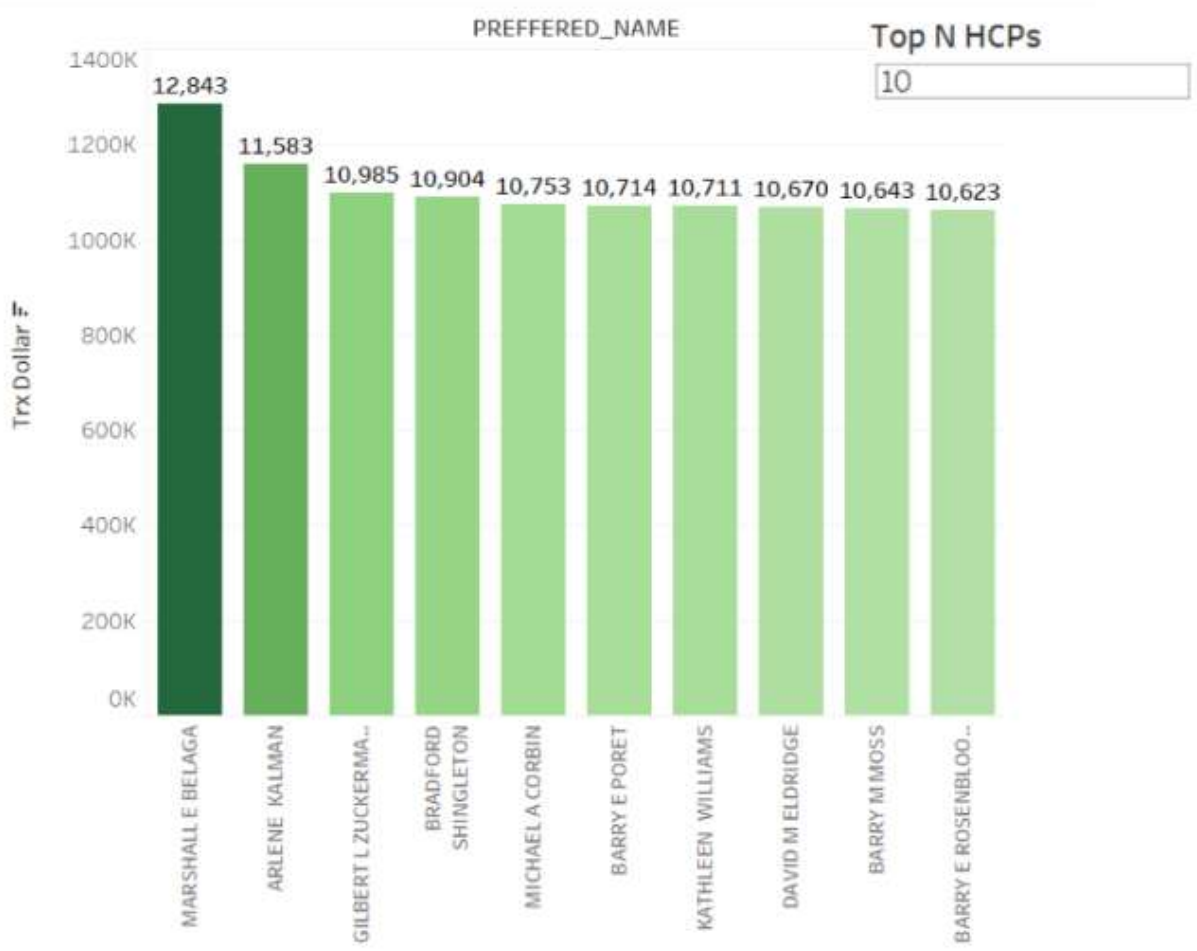


Chart insight:

- This chart depicts the top 5 territories for each market.
- The market share is decided basis sales which is decided basis(TRX Dollar)

Graph 8

Top 10 HCPs



Graph 9

CHAPTER-6

CONCLUSION

6.1 Conclusion

- The top product is 'XOPENEX 350mg'.
- The analyzed product with the highest market share is ‘ATRIPXEN 350mg’.
- The analyzed product with lowest market share is ‘PROXEN 350mg’.
- The analyzed product with the highest market share is ‘XOPENEX 350mg’ with a market share of 10.04%.
- The analyzed product with lowest market share is ‘DISPRINEX 350mg’ with a market share of 9.93%.
- The product with the highest market share is ‘CLOPIGEL 500mg’ with 7.16% market share.
- The product with lowest market share is ‘BRUFEN 350mg’ with 7.10% market share.
- The market share is decided basis sales which is decided basis(TRX Dollar)
- The territory with highest sales is Los Angles South, CA

6.2 Improvements and future work

Areas with scopes of Improvement:

Implementation of Recomputation which is self maintainable

Advantages :

- The warehouse viewing functions are completely separated from the OLTP functions.
- A discounted source will not block the process of adjusting the view of the warehouse.

Disadvantages:

- It is complex to implement
- Data is duplicated in the database.
- We need added data storage to get duplicate data.
- We are required to transfer data from sources to data storage.
- We need to utilize and maintain data transfer processes.

Implementation of Incremental Maintenance which is self maintainable

Advantages :

- The tasks of archiving the view of the database are completely separated from the activities of the OLTP.
- A discounted source will not block the process of adjusting the view of the warehouse.
- In extreme cases, the number of lines reached to maintain the view is very low.

Disadvantages:

- It is complex to implement.
- Data is duplicated in the database.
- You need additional data storage to get duplicate data.
- We are required to transfer data from sources to data storage.
- We need to utilize and maintain data transfer processes.

All methods of adjusting the view of the warehouse can be divided into mainly four categories.
They are:

- self-sustaining,
- not self-sustaining revenge
- increasing self-sustaining, and
- not self-sustaining care.

Both retrieval methods and self-care methods completely separate the functions of data repositioning functions and OLTP functions. Therefore, the correction functions will not use local data sources. These functions only use DW resources. The warehouse viewing process may continue to work in case remote data sources are not available. However, little or all of the source data is duplicated in the database in order to process the archive view. This duplicated data takes up a lot of space. Therefore we have data transfer processes which can be used to transfer data from remote data sources to our particular data warehouse. Designing, implementing and maintaining all this is time consuming. However, these are decisions, most large companies should take if they want to differentiate their data repository functions with their OLTP services. Both self-regulation and non-self-regulation have a common disability. Since remote data sources should have to process queries from a website that uses its limited resources ,this will therefore make the OLTP system to slow down.

In case the data source is not available, It will not be able to respond to queries sent from the site in desired time span. It will prevent the process of adjusting the view of the warehouse. The ever-expanding source of care has its drawbacks. To avoid an anomaly problem, the viewing process should be carefully designed.

The data repository may issue more compensation questions if multiple updates occur in data sources.
The two methods, both of them also have some common advantages:
As there is no copy of data stored on the website, no data transfer process should be used and maintained. There is no additional storage space for duplicate data. Both approaches are good for small to medium-sized companies with their slightly busy OLTP platform programs. In all four categories, additional care is best given the space used in the database and the number of lines reached to distribute the update to the intended view generated in the database.

In future work we plan to develop automated data input fields and create pipelines that will make the process more efficient. ZS has some technologies like CCF, that are available on AWS, AZURE annd many such platform that make the data visualization processes for big data a lot more easier.
We therefore plan to learn these technologies in the coming time.

REFERENCES

Meister, J., Rohde, M., Appelrath, H. and Kamp, V., 2003. Data-Warehousing im Gesundheitswesen (Data Warehousing in Health Care). *it - Information Technology*, 45(4), pp.179-185.

Christensen, J., 2017. Effective Data Visualization: The Right Chart for the Right Data, and Data Visualization: A Handbook For Data Driven Design. *Technology|Architecture + Design*, 1(2), pp.242-243.

Singh, S. and Dwivedi, D., 2020. Data Mining: Dirty Data and Data Cleaning. *SSRN Electronic Journal*,.

Kunnavil, R., 2018. Healthcare Data Utilization for the Betterment of Mankind - An Overview of Big Data Concept in Healthcare. *International Journal of Healthcare Education & Medical Informatics*, 05(02), pp.14-17.

Hussain, A. and Roy, A., 2016. The emerging era of Big Data Analytics. *Big Data Analytics*, 1(1).