PHARMALYTICA

Pharmalytica

Team 3

Mentor: Ziad Mohamed Saad

`

Pharmalytica

Content:

`

Pharmalytica

Business Analyst:

Overview:

Project Overview and Background:

This project aims to enhance pharmacy operations by developing a smart, data-driven dashboard that focuses on tracking drug sales, product performance, and supplier contribution. The system will enable pharmacy stakeholders to make informed decisions based on accurate, real-time data.

Key Benefits:

1. Monitor sales by brand, dosage form, and active ingredients
2. Suggest alternative drugs by replacing brand names with active ingredients
3. Identify fast-moving and underperforming products
4. Improve supplier visibility and contribution analysis

Data Sources:

- ERP Oracle / Kaggle / by python / WHO

Benefits for Business:

1. Instant visibility into drug sales by brand, dosage form, and active ingredients
2. Identification of fast- and slow-moving drugs to identify the sales of each product
3. Intelligent substitution support by analyzing active ingredients for alternative drug suggestions
4. Supplier performance tracking based on delivery accuracy and consistency
5. Real-time dashboards and automated reporting for quick, informed decision-making
6. Monthly and yearly sales trend analysis to improve forecasting and promotional planning
7. Anomaly detection, such as sudden drops in sales or high return rates
8. KPI-driven insights to enhance strategic, data-backed decision-making across the organization

`

Pharmalytica

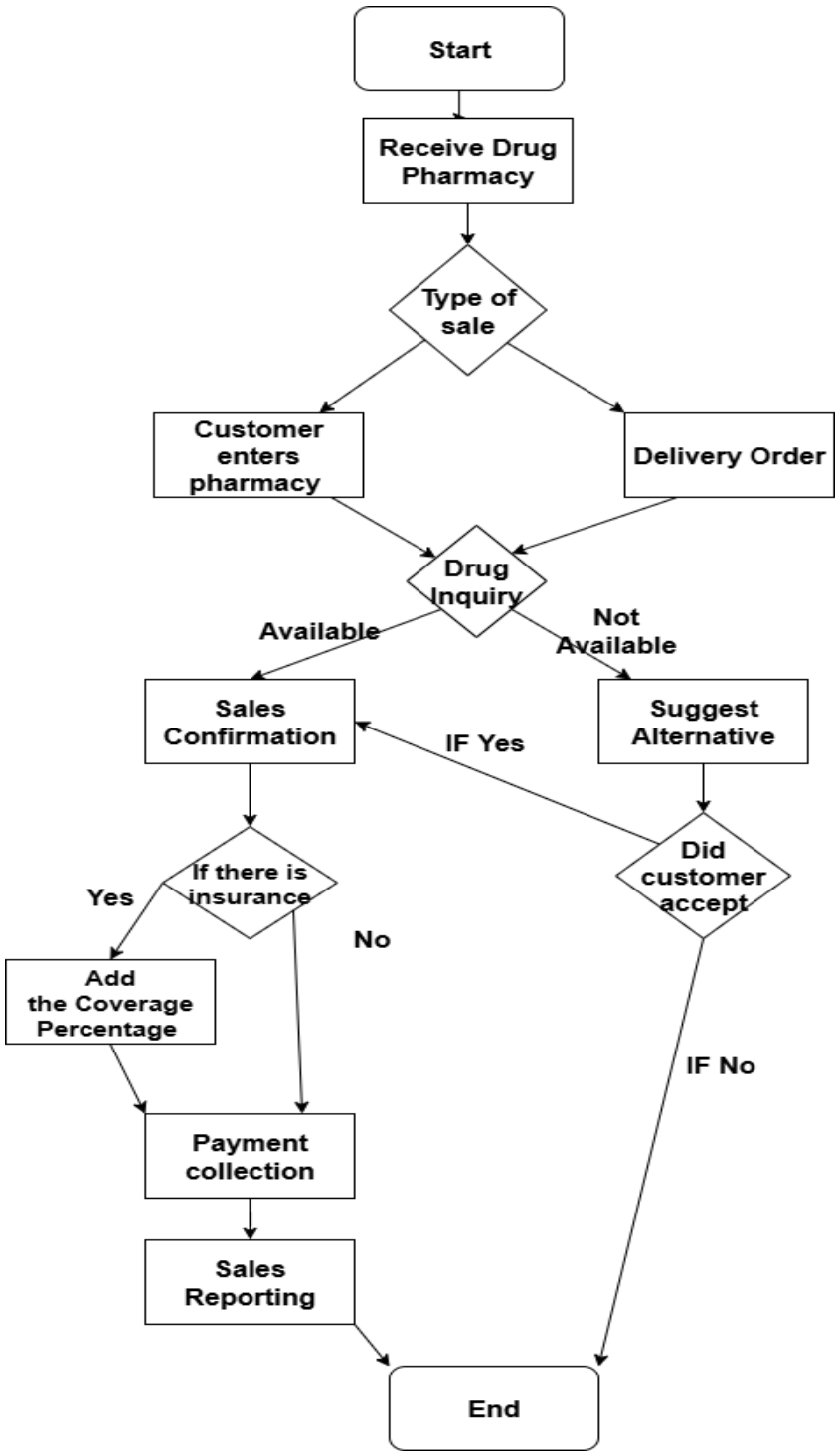
Key Assumptions:

#	Assumptions
1.	**Centralized Product Catalog Assumption:** It is assumed that all pharmacies will follow a standardized product catalog to ensure consistent naming, categorization, and tracking of products. This allows for accurate reporting and prevents duplication across different branches. Any inconsistencies will be resolved during the initial data entry phase.
2.	**Sales Data Availability Assumption:** It is assumed that pharmacies will provide complete sales data, either through direct entry or scheduled uploads. This data is essential for monitoring sales performance and generating accurate reports.
3.	**Multi-Pharmacy Integration Assumption:** The system is assumed to support integration with multiple pharmacies across different zones, with each assigned a unique identifier. Sales and product data will be consolidated to allow unified reporting and performance analysis. A consistent data format will be used to ensure smooth integration.
4.	**Product Taxonomy Assumption:** It is assumed that all products will be categorized based on their active ingredients, with commercial names included to avoid duplication. This standard taxonomy will support accurate sales comparison, improve searchability, and maintain data integrity across the system.
5.	**Access to KPI Information Assumption:** It is assumed that the pharmacy management will provide access to relevant documents, data, and key performance indicators to support the IT team. This access will enable the team to capture accurate information related to the targeted functional areas of the system.
6.	**Timely Data Updates:** It is assumed that all pharmacies will provide accurate, standardized sales data, including product, dosage, and manufacturer details. Data will be updated regularly—ideally daily—to ensure timely reporting and reliable KPIs. The project also expects smooth integration with existing systems and compliance with all relevant regulations.

`

Business Requirements:

High Level Procurement Process Flow:



`

Pharmalytica

Business terms and definitions:

Business Terms	Business Definitions
product_id	Unique identifier for each product.
brand_name	Commercial name under which the product is sold.
clean_brand_name	Standardized version of the product brand name for consistency in reporting.
dosage_form	The physical form in which the drug is administered (e.g., tablet, syrup).
pack_size	Quantity of dosage units contained in one pack.
pack_unit	The unit in which packs are counted (e.g., box, blister, bottle).
product_type	Categorization of product as “drug” or “supply”.
price_inr	Selling price of the product in Indian Rupees (or system currency).
invoice	Unique identifier for a sale transaction.
sales_sheet	Number of units sold measured in sheet format (e.g., 1 sheet of 10 tablets).
sales_pack	Number of packs sold in the transaction.
sheet	Units contained in one sheet.
added_date	Date the transaction or product was added.
sale_time	Timestamp of the sale.
sales_type	Type of sale (e.g., in-pharmacy, delivery)
group_key	Composite identifier for pharmacy location (e.g., city + zone + pharmacy ID).
pharmacy_id	Unique identifier of the pharmacy outlet.
zone_id	Code representing the regional zone.
city_id	Code representing the city.
manufacturer_id	Unique ID for each drug manufacturer.
manufacturer_name	Name of the manufacturer producing the drug.
ingredient_id	Unique identifier for each active ingredient.
ingredient_name	Name of the primary chemical ingredient.
therapeutic_class	Classification based on treatment purpose (e.g., analgesic, antibiotic).
insurance_id	Unique ID of the insurance company or policy.
insurance_name	Name of the insurance provider.
average_coverage	Average percentage of cost covered by the insurance plan.

`

Pharmalytica

Defined Key KPI's/Metrics/Measures:

Sales Dashboard:

Measures	Description	Condition
Invoice Revenue	Total number of unique invoices issued	COUNTD(Invoice) or DISTINCTCOUNT(Invoice)
Avg Sales/Day	Average total sales per day	SUM(Sales_Sheet × Sheet) ÷ Number of Days
Total Invoices	Number of invoices	DISTINCTCOUNT('sales'[invoic e_bk])
Total Sold Units	Total number of units sold across all products	SUM(Sales_Sheet × Sheet)
Revenue Per Day	Trend of revenue generated per day	SUM(Revenue) GROUP BY Added_Date
Therapeutic Class Revenue	Revenue breakdown by therapeutic class	SUM(Revenue) GROUP BY Therapeutic_Class
Revenue Per Sale Type	Share of revenue by sale type (delivery vs in-pharmacy)	SUM(Revenue) GROUP BY Is_Delivery
Amount of Each Product Sold	Volume sold per product	SUM(Sales_Sheet × Sheet) GROUP BY Product_Name
Top 3 Manufacturer Revenue	Top contributing manufacturers by revenue	TOPN(3, SUM(Revenue)) GROUP BY Manufacturer
Total Measures	9	

`

Product Dashboard:

Measures	Description	Condition
Number of Ingredients	Total count of distinct primary ingredients	COUNT(DISTINCT Primary_Ingredient)
Number of Unique Products Sold	Count of distinct product names sold	COUNT(DISTINCT Product_Name)
Top Manufacturers by Dosage Form	Most selling manufacturers for each dosage form	SUM(Sales) GROUP BY Manufacturer, Dosage_Form
Top 3 Most Profitable Ingredients	Ingredients with the highest revenue	TOPN(3, Primary_Ingredient, SUM(Profit))
Top 5 Highest Selling Products	Products with the highest total units sold	TOPN(5, Product_Name, SUM(Sales_Units))
Top 5 Ingredients by Unit Sales	Ingredients that contribute most to sales volume	TOPN(5, Primary_Ingredient, SUM(Sales_Units))
Top 5 Sales by Dosage Form	Breakdown of total units sold by dosage form	SUM(Sales_Units) GROUP BY Dosage_Form
Total Measures	7	

`

Time Dashboard:

Measures	Description	Condition
Avg Sales/Qtr	Average revenue per quarter	SUM(Revenue) ÷ 4
Peak Hour Sales	Hour of the day with the highest revenue	MAX(SUM(Revenue)) GROUP BY Hour(Time)
Top Revenue Day	Day of week with highest revenue	MAX(SUM(Revenue)) GROUP BY Weekday
Sold Units/Day	Average number of sold units per day	SUM(Sales Sheet × Sheet) ÷ Number of Days
Sold Units by City and Quarter	Comparison of sold units across cities for each quarter	SUM(Units) GROUP BY City, Quarter
Top Daily-Selling Manufacturer	Manufacturer with highest total daily sales	MAX(SUM(Revenue)) GROUP BY Manufacturer
Revenue Over Date	Revenue aggregated by quarter	SUM(Revenue) GROUP BY Quarter
Weekdays Sales	Revenue share by day of week	SUM(Revenue) GROUP BY Weekday
Revenue Per Time	Sales revenue distribution over hours of the day	SUM(Revenue) GROUP BY Time (hour)
Total Measures	9	

`

Location Dashboard:

Measures	Description	Condition
Avg Rev/pharmacy	Average revenue generated per pharmacy across all cities	SUM(Revenue) ÷ COUNT(DISTINCT Pharmacy_ID)
Top City	City with highest total revenue	MAX(SUM(Revenue)) GROUP BY City
Total Pharmacies	Number of unique pharmacies across all cities	COUNT(DISTINCT Pharmacy_ID)
Top Manufacturer per City	Distribution of sales per manufacturer per city	SUM(Revenue) GROUP BY City, Manufacturer
Top 4 Revenue	The 4 highest revenue-generating product-location combinations	TOPN(4, Product × Pharmacy, SUM(Revenue))
Revenue by Month and City	Trend of revenue per city across months	SUM(Revenue) GROUP BY Month, City
Sales Per City	Distribution of total sales volume per city	SUM(Sales Units) GROUP BY City
City Revenue Map	Geographic distribution of revenue by city	MAP(City, SUM(Revenue))
Total Measures	8	

`

Delivery Dashboard:

Measures	Description	Condition
Avg Sales per Delivery Transaction	Average revenue per delivery invoice	SUM(Price × Units) ÷ COUNT(DISTINCT Invoice) WHERE Is_Delivery = TRUE
Number of Each Pick-up Type	Total number of transactions for delivery and on-site modes	COUNT(*) GROUP BY Pickup_Type
Total Revenue (Delivery & On-site)	Total revenue by transaction type	SUM(Sales WHERE Is_Delivery = TRUE) and SUM(Sales WHERE Is_Delivery = FALSE)
Pick-up Form per Time	Count of pick-ups for each month and pickup method	COUNT(Pickups) GROUP BY Month, Pickup_Type
Revenue by Type and Channel	Revenue split by product type (drug/supply) and transaction type	SUM(Price × Quantity) GROUP BY Product_Type, Is_Delivery
Top 5 Classes Sold via Delivery	Top therapeutic classes by sales through delivery channel	TOPN(5, Therapeutic_Class, SUM(Sales) WHERE Is_Delivery = TRUE)
Top 5 Manufacturers Relying on Delivery	Manufacturers with highest reliance on delivery channel for sales	SUM(Sales WHERE Is_Delivery = TRUE) ÷ SUM(All Sales) GROUP BY Manufacturer
Total Measures	7	

`

Pharmalytica

Insurance Dashboard:

Measures	Description	Condition
Insurance Sales Trend	Monthly trend of total insurance revenue and number of insurance transactions	SUM(Price) and COUNTD(Invoice Bk) grouped by MONTH(Date Sk) filtered by Insurance Name
Insurance Sales by City & Zone	Insurance revenue split across cities and zones	SUM(Price) grouped by City (copy) and Zone (copy) filtered by Insurance Name
Top 5 Insured Products Sold	Products with the highest number of insured invoices	TOPN(5) of Brand Name by COUNTD(Invoice Bk) filtered by Insurance Name
Insurance Sales by Dosage Form	Sales share of insured products based on dosage form	Pie chart using COUNTD(Invoice Bk) grouped by Dosage Form filtered by Insurance Name
Total Insurance Units Sold	Total number of insurance transactions (proxy for units)	COUNTD(Invoice Bk) filtered by Insurance Name
Total Measures	5	

`

Summary Dashboard:

Measures	Description	Condition
Top Frequent City	Most frequent city by transactions	MODE(City)
Unique Cities Count	Number of distinct cities	COUNTDISTINCT(City)
Total Manufacturers	Number of manufacturers	COUNTDISTINCT(Manufacturer)
Top Frequent Manufacturer	Most common manufacturer across records	MODE(Manufacturer)
Top Revenue Day	Day of week with highest revenue	TOP1(Day ORDER BY Revenue DESC)
Total Revenue	Total revenue from all transactions	SUM(Revenue)
Total Sold Units	Sum of sold units	SUM(Sold_Units)
Top-Selling Product	Product with highest unit sales	TOP1(Product ORDER BY Units_Sold DESC)
Top Selling Hours	Peak sales hours	TOP N(Hour ORDER BY Revenue DESC)
Sales by Therapeutic Class	Distribution of sales per therapeutic class	SUM(Sales) GROUP BY Therapeutic_Class
Sales Per City	Distribution of sales per city	SUM(Sales) GROUP BY City
Total Measures	11	

`

Detailed Dashboard:

Measures	Description	Condition
Sum of Sold Units	Total quantity of units sold across all products	SUM(Sold_Units)
Total Revenue	Total revenue across all products	SUM(Total_Revenue)
Sales by Dosage Form	Sales amount broken down by dosage type	SUM(Sales) GROUP BY Dosage_Form
Sales Per City	Total sales by location (e.g., Cairo, Alexandria, Port Said)	SUM(Sales) GROUP BY City
Sales by Month	Monthly revenue trends across the year	SUM(Sales) GROUP BY Month
Invoices by Hours	Total invoice value distributed by hour of day	SUM(Invoice_Amount) GROUP BY Hour
Sales by Sales Type	Revenue split between on-site and delivery sales	SUM(Sales) GROUP BY Sales_Type (On-site/Delivery)
0Total Measures	7	

`

Pharmalytica

Target Audience:

IDC	
Power Users	Business Owner
Beta users	Pharmacy Operating team
End users	Pharmacy Management
Concurrent users	9
Data Update frequency	Data Update frequency

Filters:

#	Filter Name	Source Table Name /IT Team	Description
1	Quarter		The three-month period within a financial or calendar year during which sales and performance are tracked (e.g., Q1, Q2, Q3, Q4).
2	Month		The calendar month when the sales transaction or event occurred.
3	Weeks		The week number or date range within a month for detailed time-based analysis of sales and activities.
4	Pharmacy		The specific pharmacy branch or outlet where the transaction took place.
5	Zone		The geographic area or operational region grouping multiple pharmacies within a city.
6	City		The city in which the pharmacy is located.
7	Dosage form		The physical form of the medication sold (e.g., tablet, syrup, injection).
8	Product Type		Indicates whether the item is a drug (pharmaceutical product) or a supply (non-drug item such as medical equipment, consumables, or accessories).
9	Sale Type		Indicates whether the item is a delivery or on site.
10	Active ingredient		The main chemical compound in the medication responsible for its therapeutic effect.

`

Data Modelers (Yassin Elmaghrabi – Ahmed Ibrahim):

Work Completed

Data Modeling & Databases

- Designed and implemented **production schema** and **production database**. (Yassin And Ahmed)
- Built **data warehouse schema** for analytics and reporting. (Yassin And Ahmed)
- Created **entity-relationship diagram (ERD)** to capture system design. (Yassin And Ahmed)
- Developed **mapping sheet** for consistent data transformation and loading. (Yassin And Ahmed)
- Augmented datasets by adding **ingredients, active ingredients, therapeutic classes, and pricing**. (Yassin)
- Automated processing of source CSV into multiple smaller structured files.

Backend & Data Science

5. Implemented a **Python backend** with: (Yassin)
    a. **Clustering algorithms** for data segmentation.
    b. **Association rules mining** for discovering relationships in pharmaceutical data.
    c. A **retrieval-augmented generation (RAG) chatbot** for intelligent Q&A.
6. Integrated **Redis database** for caching and fast retrieval. (Yassin)
7. Used **Docker Compose** to orchestrate all application components. (Yassin)

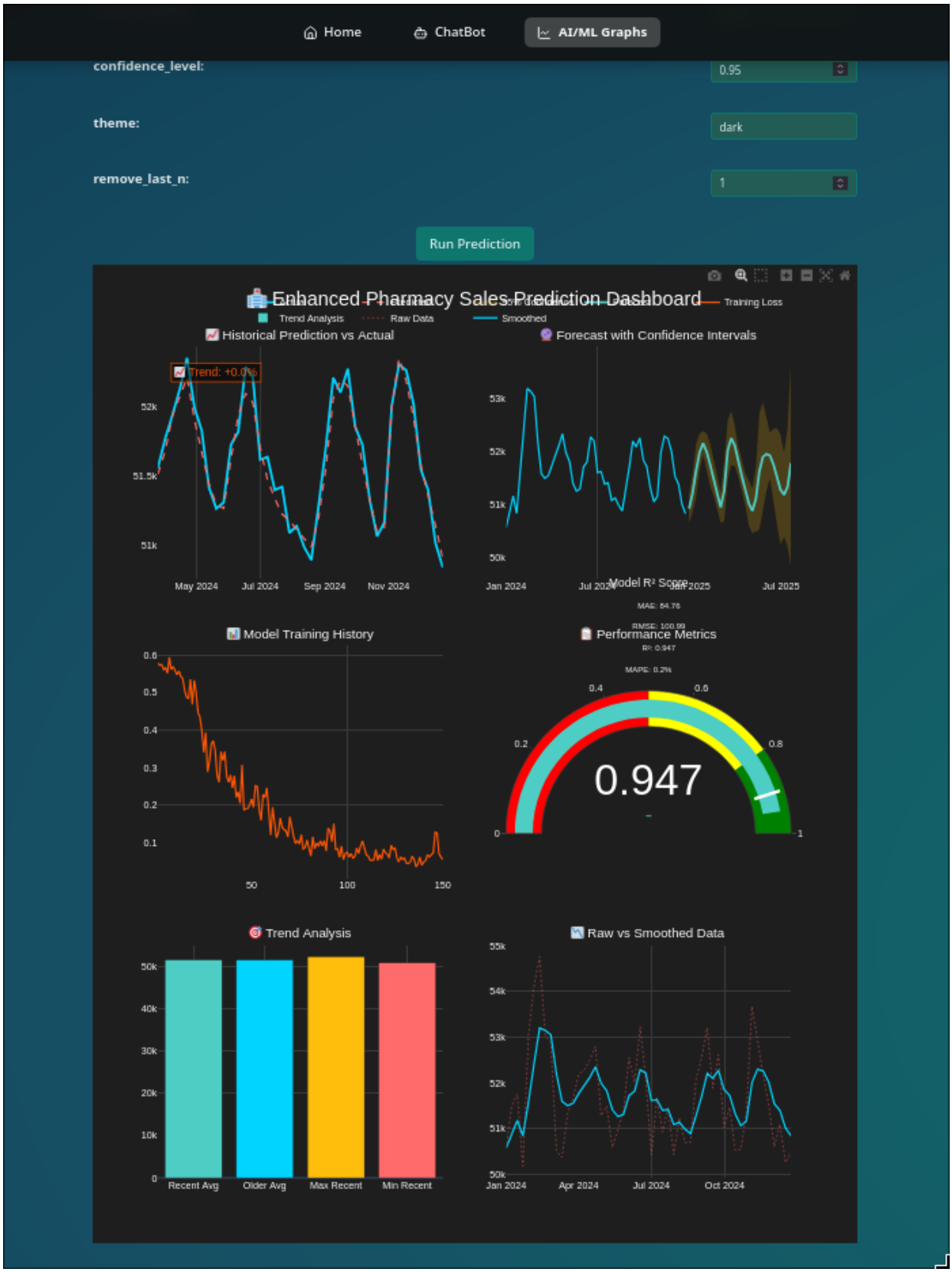
Frontend & User Experience

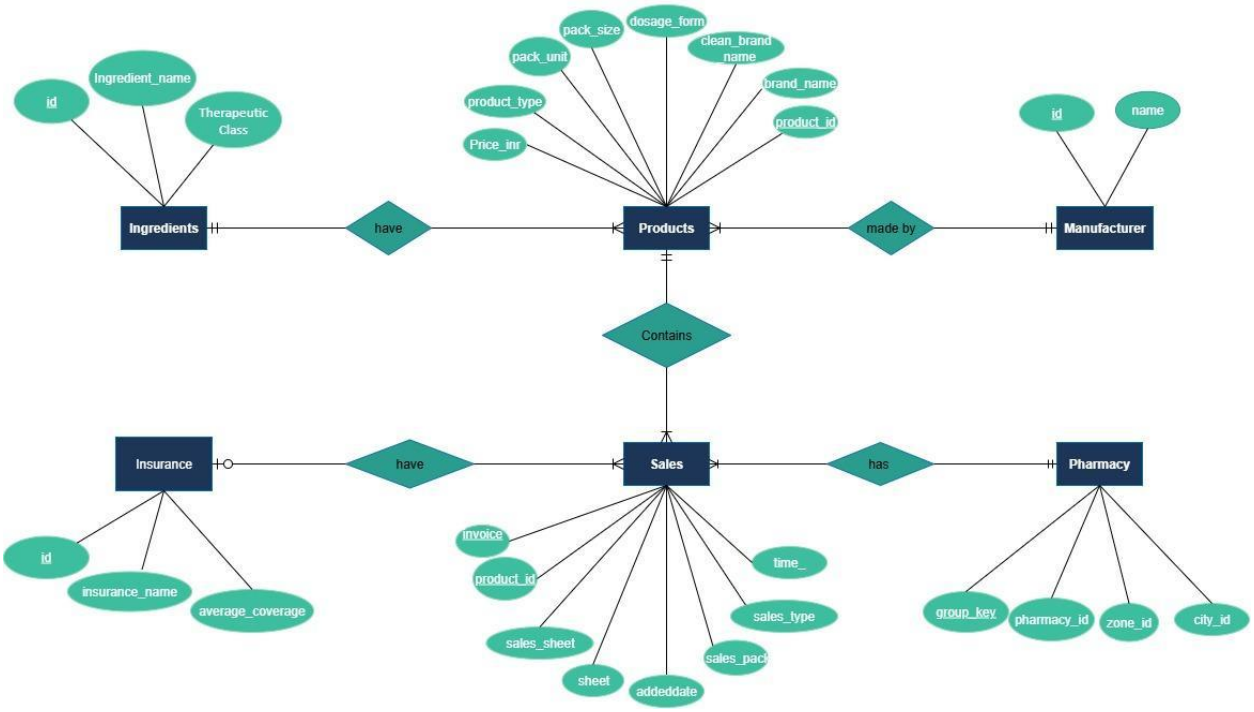
- Built a **Svelte frontend** to expose AI tools and the chatbot through a user-friendly interface. (Yassin)

Deliverables

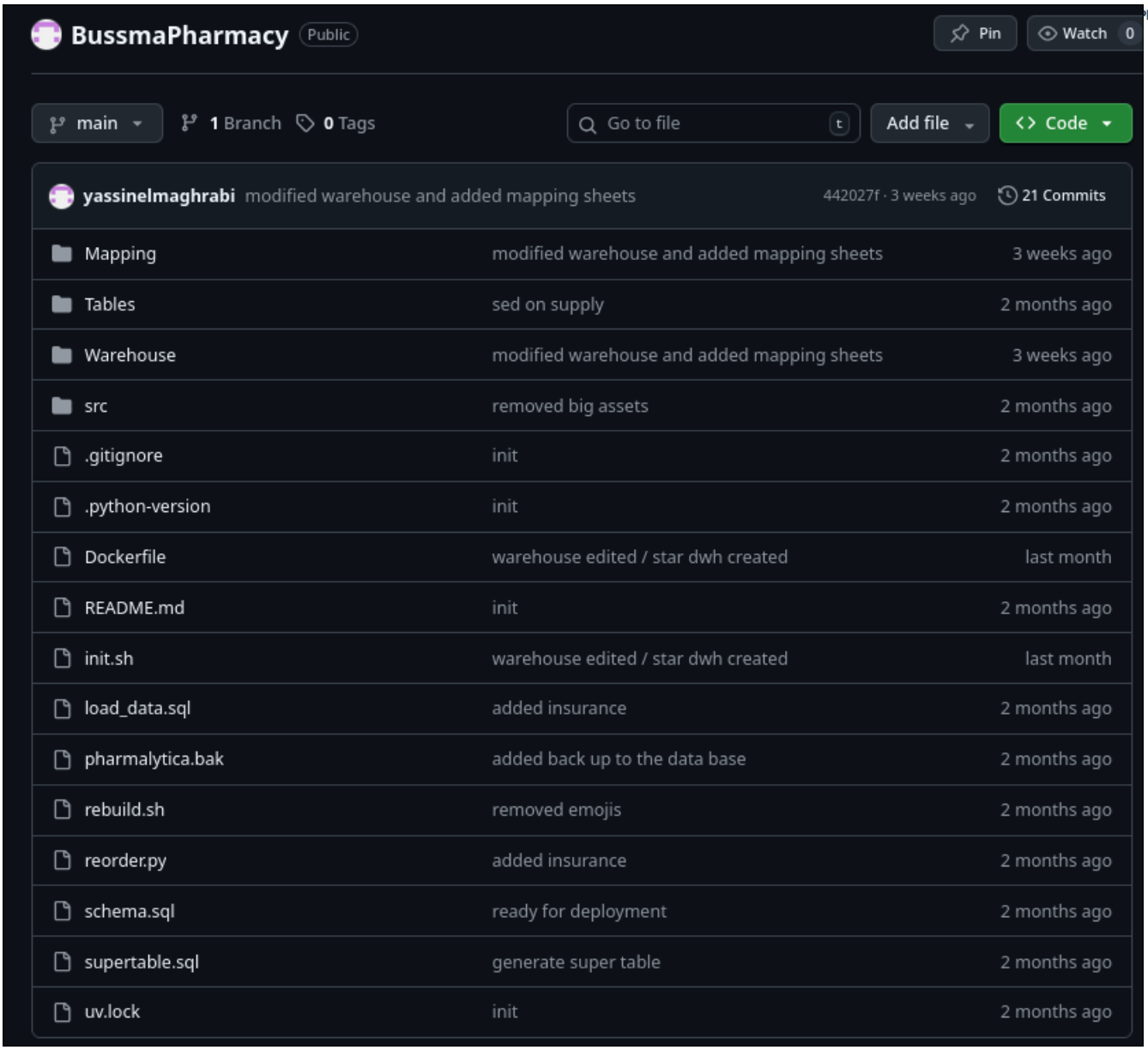
- Production-ready database schema and warehouse design.
- ERD and mapping documentation.
- End-to-end AI/ML application with chatbot interface.
- Reproducible deployment using Docker Compose.

`


Home
ChatBot
AI/ML Graphs
confidence_level:
0.95
theme:
dark
remove_last_n:
1
Run Prediction
Enhanced Pharmacy Sales Prediction Dashboard
Training Loss
Trend Analysis
Raw Data
Smoothed
Historical Prediction vs Actual
Trend: +0.0%
Forecast with Confidence Intervals
Model R² Score
MAE: 84.76
RMSE: 100.99
Model Training History
Performance Metrics
R²: 0.947
MAPE: 0.2%
0.947
Trend Analysis
Recent Avg
Older Avg
Max Recent
Min Recent
Raw vs Smoothed Data

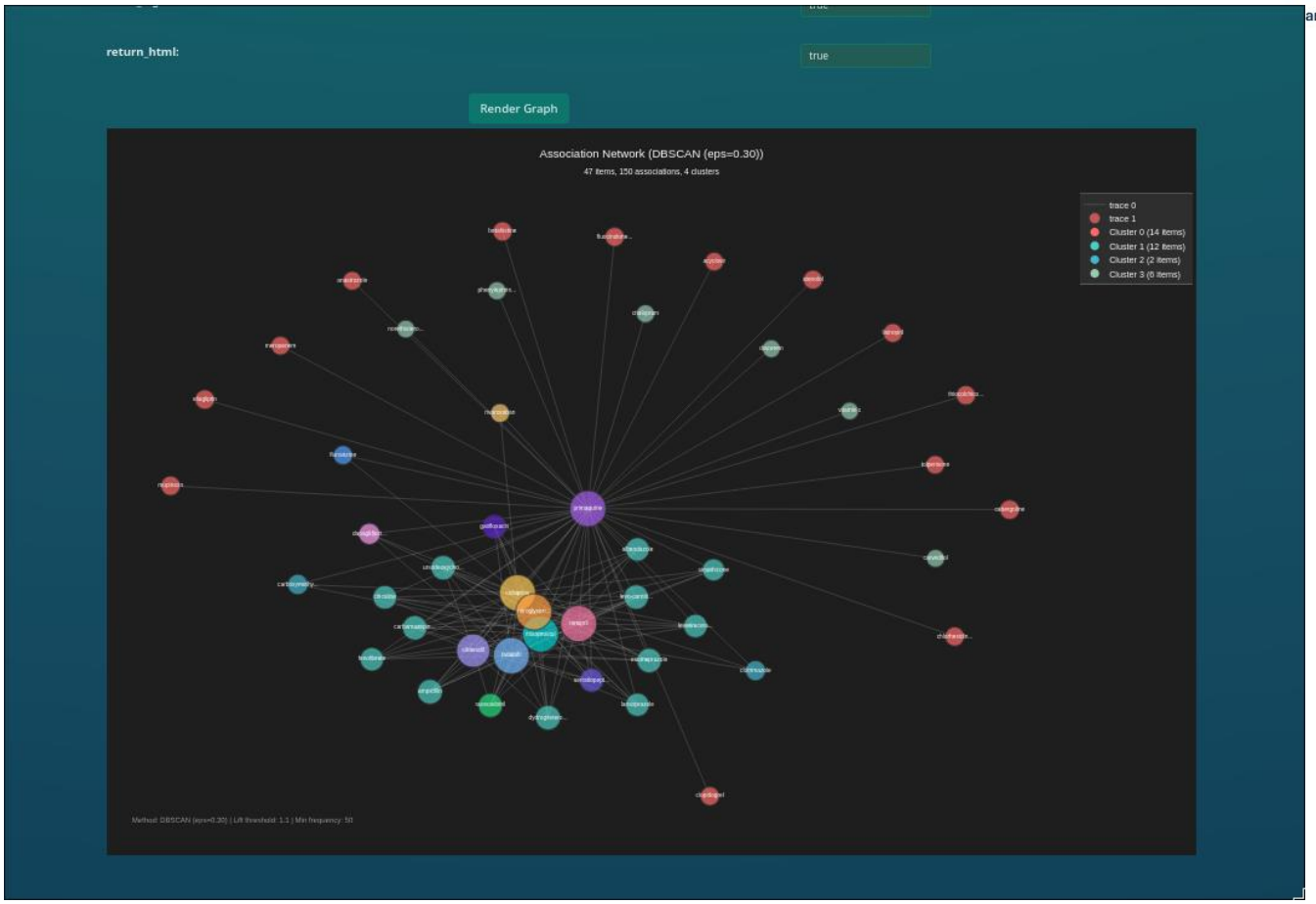

pack_size
dosage_form
pack_unit
clean_brand name
brand_name
product_type
product_id
Price_inr
id
Ingredient_name
Therapeutic Class
id
name
Ingredients
have
Products
made by
Manufacturer
Contains
Insurance
have
Sales
has
Pharmacy
invoice
time_
product_id
sales_type
sales_sheet
sales_pack
sheet
addeddate
id
insurance_name
average_coverage
group_key
pharmacy_id
zone_id
city_id

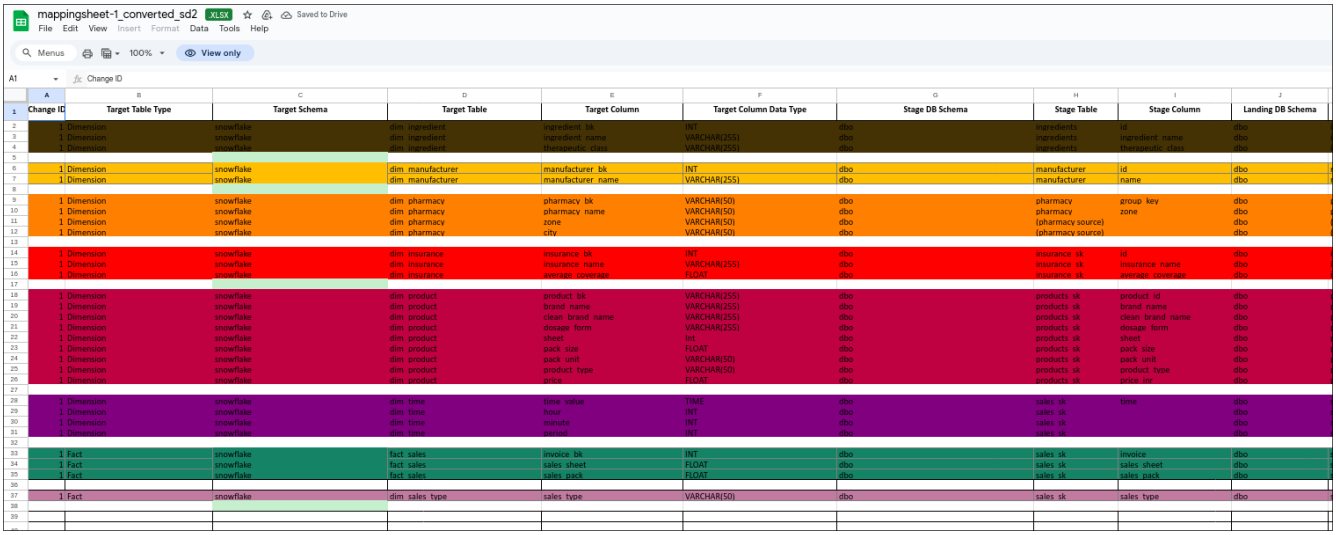
`


BussmaPharmacy
Public
Pin
Watch 0
main
1 Branch
0 Tags
Go to file
t
Add file
Code
yassinelmaghrabi modified warehouse and added mapping sheets
442027f · 3 weeks ago
21 Commits
Mapping modified warehouse and added mapping sheets 3 weeks ago
Tables sed on supply 2 months ago
Warehouse modified warehouse and added mapping sheets 3 weeks ago
src removed big assets 2 months ago
.gitignore init 2 months ago
.python-version init 2 months ago
Dockerfile warehouse edited / star dwh created last month
README.md init 2 months ago
init.sh warehouse edited / star dwh created last month
load_data.sql added insurance 2 months ago
pharmalytica.bak added back up to the data base 2 months ago
rebuild.sh removed emojis 2 months ago
reorder.py added insurance 2 months ago
schema.sql ready for deployment 2 months ago
supertable.sql generate super table 2 months ago
uv.lock init 2 months ago


yassinelmaghrabi don't remember what changed
aeb49f0 · last month
4 Commits
backend don't remember what changed last month
frontend don't remember what changed last month
.gitignore init last month
README.md readme last month
docker-compose.yml init last month

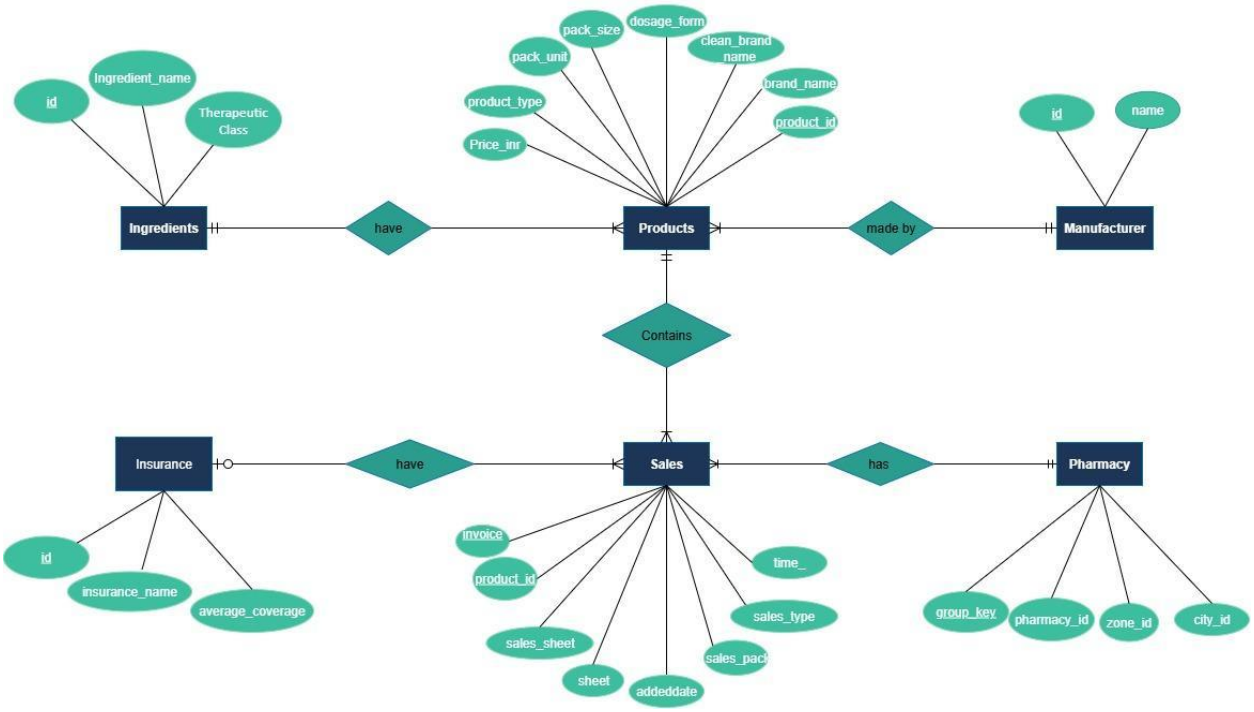
`

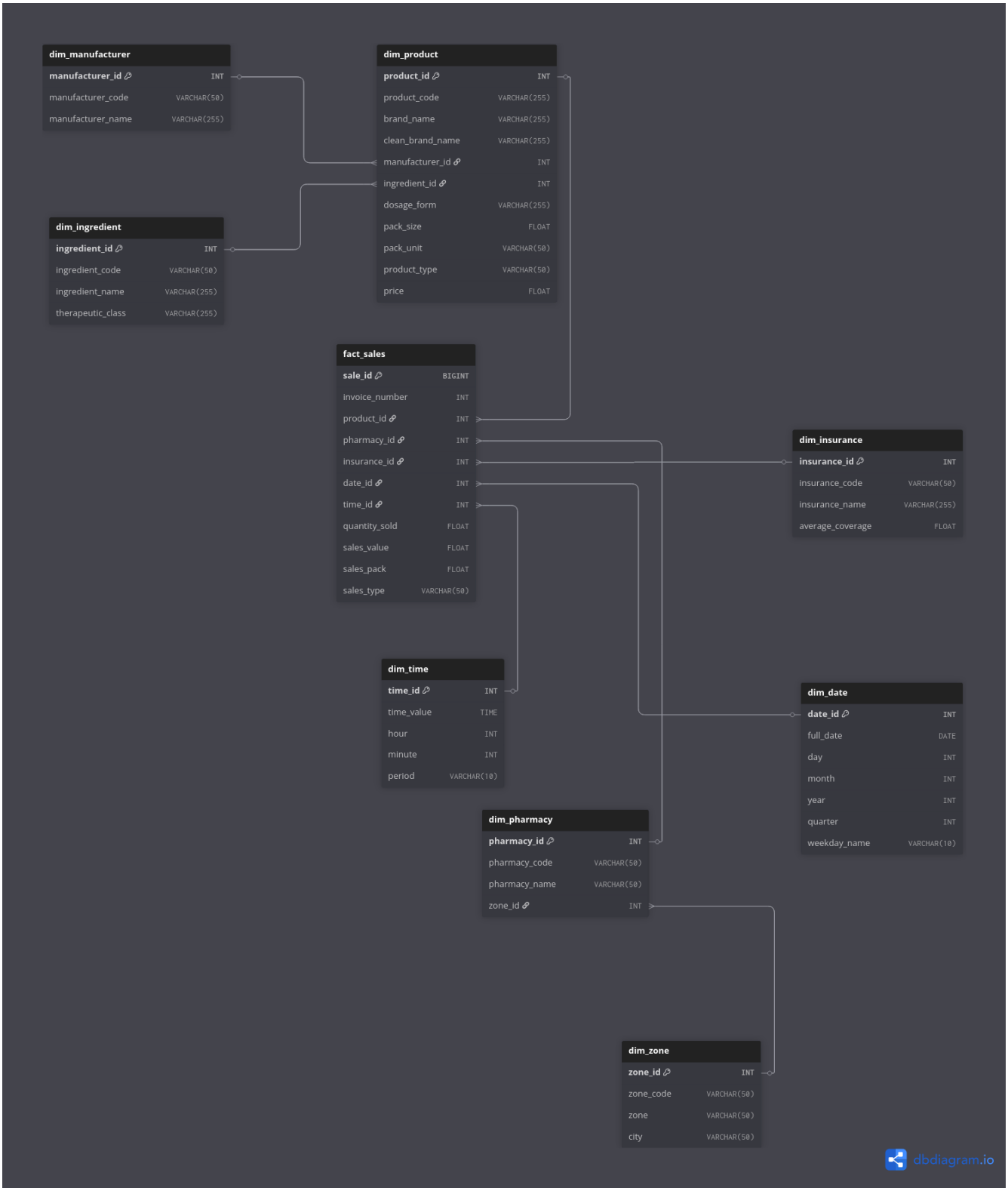




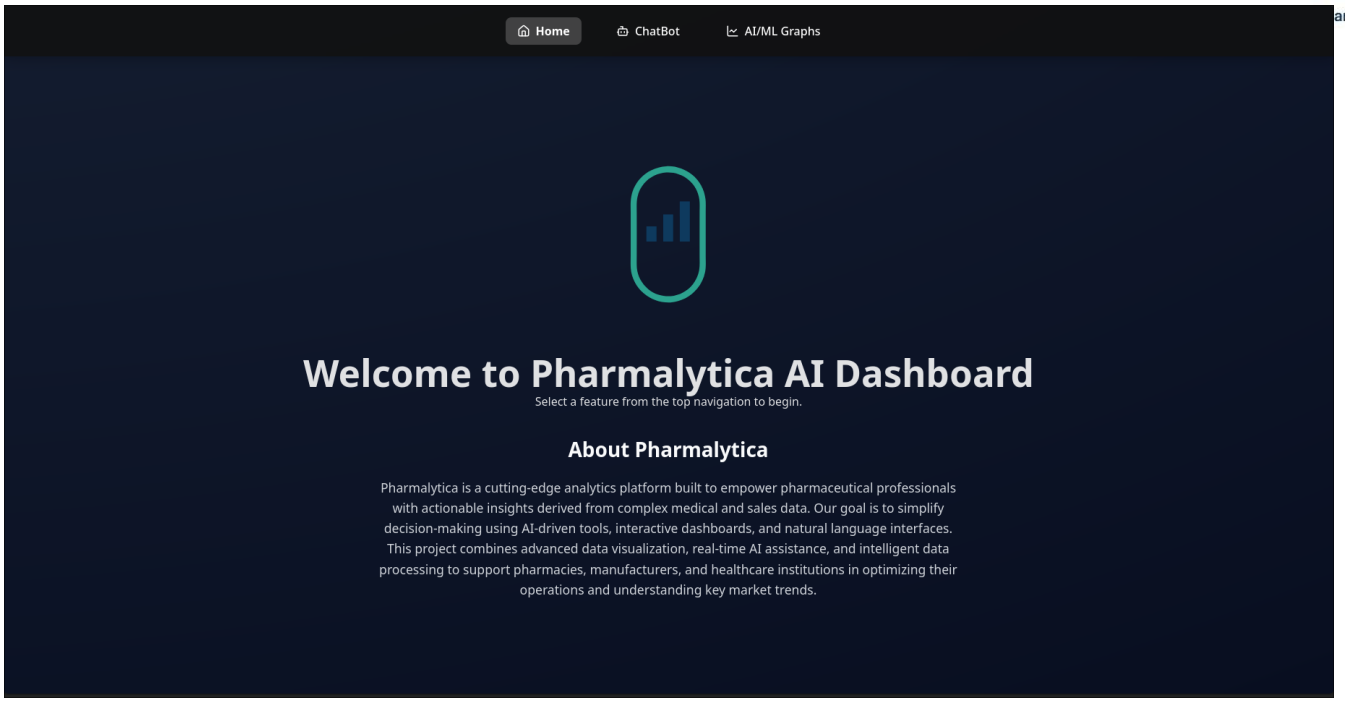
mappingsheet-1_converted_sd2 XLSX

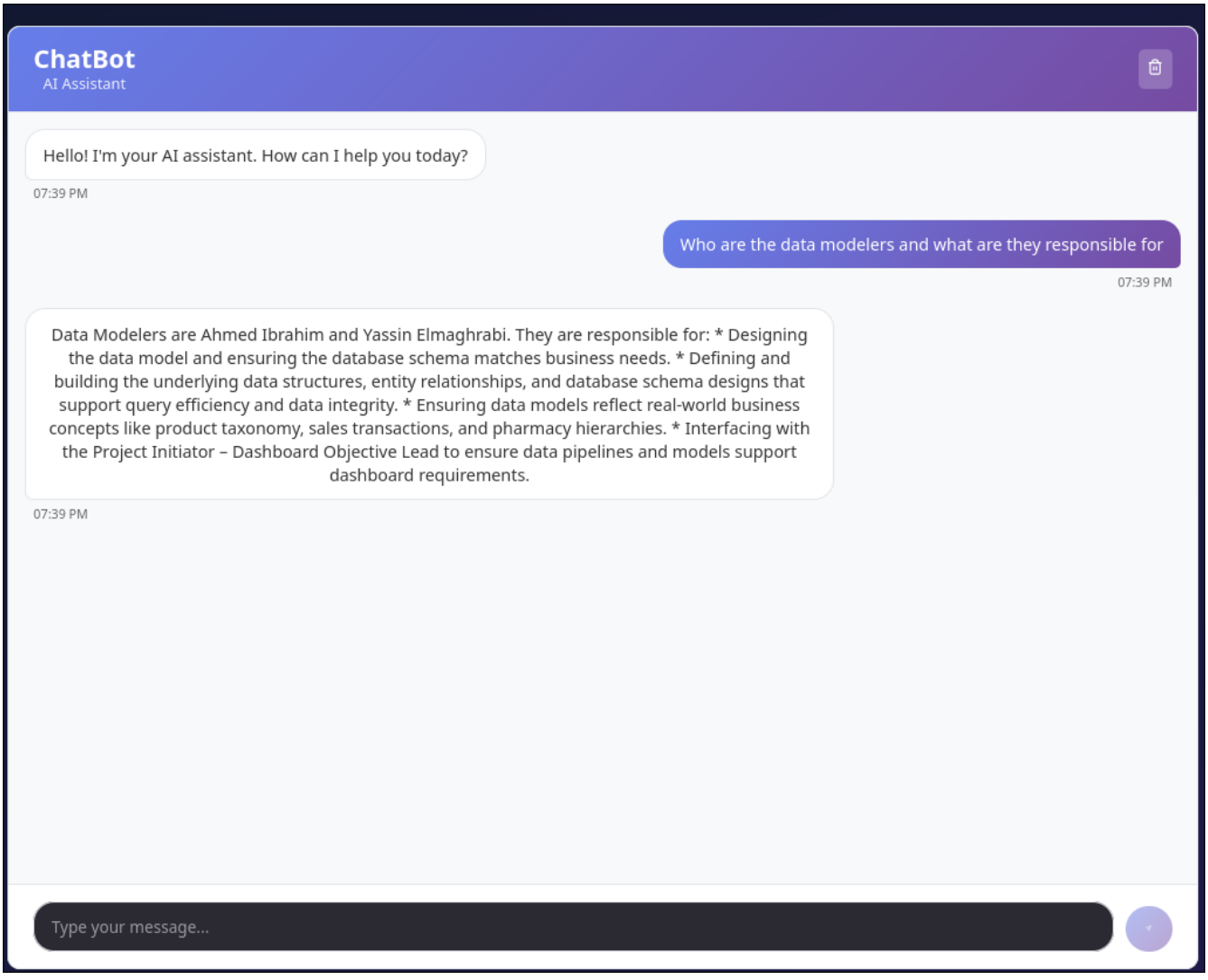
Change ID	Target Table Type	Target Schema	Target Table	Target Column	Target Column Data Type	Stage DB Schema	Stage Table	Stage Column	Landing DB Schema
1	Dimension	snowflake	dim_ingredient	ingredient_bk	INT	dbo	ingredients	id	dbo
1	Dimension	snowflake	dim_ingredient	ingredient_name	VARCHAR(255)	dbo	ingredients	ingredient_name	dbo
1	Dimension	snowflake	dim_ingredient	therapeutic_class	VARCHAR(255)	dbo	ingredients	therapeutic_class	dbo
1	Dimension	snowflake	dim_manufacturer	manufacturer_bk	INT	dbo	manufacturer	id	dbo
1	Dimension	snowflake	dim_manufacturer	manufacturer_name	VARCHAR(255)	dbo	manufacturer	name	dbo
1	Dimension	snowflake	dim_pharmacy	pharmacy_bk	VARCHAR(50)	dbo	pharmacy	group_key	dbo
1	Dimension	snowflake	dim_pharmacy	pharmacy_name	VARCHAR(50)	dbo	pharmacy	zone	dbo
1	Dimension	snowflake	dim_pharmacy	zone	VARCHAR(50)	dbo	(pharmacy source)		dbo
1	Dimension	snowflake	dim_pharmacy	city	VARCHAR(50)	dbo	(pharmacy source)		dbo
1	Dimension	snowflake	dim_insurance	insurance_bk	INT	dbo	insurance_sk	id	dbo
1	Dimension	snowflake	dim_insurance	insurance_name	VARCHAR(255)	dbo	insurance_sk	insurance_name	dbo
1	Dimension	snowflake	dim_insurance	average_coverage	FLOAT	dbo	insurance_sk	average_coverage	dbo
1	Dimension	snowflake	dim_product	product_bk	VARCHAR(255)	dbo	products_sk	product_id	dbo
1	Dimension	snowflake	dim_product	brand_name	VARCHAR(255)	dbo	products_sk	brand_name	dbo
1	Dimension	snowflake	dim_product	clean_brand_name	VARCHAR(255)	dbo	products_sk	clean_brand_name	dbo
1	Dimension	snowflake	dim_product	dosage_form	VARCHAR(255)	dbo	products_sk	dosage_form	dbo
1	Dimension	snowflake	dim_product	sheet	int	dbo	products_sk	sheet	dbo
1	Dimension	snowflake	dim_product	pack_size	FLOAT	dbo	products_sk	pack_size	dbo
1	Dimension	snowflake	dim_product	pack_unit	VARCHAR(50)	dbo	products_sk	pack_unit	dbo
1	Dimension	snowflake	dim_product	product_type	VARCHAR(50)	dbo	products_sk	product_type	dbo
1	Dimension	snowflake	dim_product	price	FLOAT	dbo	products_sk	price_lnr	dbo
1	Dimension	snowflake	dim_time	time_value	TIME	dbo	sales_sk	time	dbo
1	Dimension	snowflake	dim_time	hour	INT	dbo	sales_sk		dbo
1	Dimension	snowflake	dim_time	minute	INT	dbo	sales_sk		dbo
1	Dimension	snowflake	dim_time	period	INT	dbo	sales_sk		dbo
1	Fact	snowflake	fact_sales	invoice_bk	INT	dbo	sales_sk	invoice	dbo
1	Fact	snowflake	fact_sales	sales_sheet	FLOAT	dbo	sales_sk	sales_sheet	dbo
1	Fact	snowflake	fact_sales	sales_pack	FLOAT	dbo	sales_sk	sales_pack	dbo
1	Fact	snowflake	dim_sales_type	sales_type	VARCHAR(50)	dbo	sales_sk	sales_type	dbo


id
Ingredient_name
Therapeutic Class
Ingredients
have
pack_size
dosage_form
pack_unit
clean_brand name
brand_name
product_type
product_id
Price_inr
Products
made by
id
name
Manufacturer
Contains
Insurance
have
Sales
has
Pharmacy
id
insurance_name
average_coverage
invoice
product_id
time_
sales_type
sales_sheet
sales_pack
sheet
addeddate
group_key
pharmacy_id
zone_id
city_id


dim_manufacturer
manufacturer_id INT
manufacturer_code VARCHAR(50)
manufacturer_name VARCHAR(255)
dim_product
product_id INT
product_code VARCHAR(255)
brand_name VARCHAR(255)
clean_brand_name VARCHAR(255)
manufacturer_id INT
ingredient_id INT
dosage_form VARCHAR(255)
pack_size FLOAT
pack_unit VARCHAR(50)
product_type VARCHAR(50)
price FLOAT
dim_ingredient
ingredient_id INT
ingredient_code VARCHAR(50)
ingredient_name VARCHAR(255)
therapeutic_class VARCHAR(255)
fact_sales
sale_id BIGINT
invoice_number INT
product_id INT
pharmacy_id INT
insurance_id INT
date_id INT
time_id INT
quantity_sold FLOAT
sales_value FLOAT
sales_pack FLOAT
sales_type VARCHAR(50)
dim_insurance
insurance_id INT
insurance_code VARCHAR(50)
insurance_name VARCHAR(255)
average_coverage FLOAT
dim_time
time_id INT
time_value TIME
hour INT
minute INT
period VARCHAR(10)
dim_date
date_id INT
full_date DATE
day INT
month INT
year INT
quarter INT
weekday_name VARCHAR(10)
dim_pharmacy
pharmacy_id INT
pharmacy_code VARCHAR(50)
pharmacy_name VARCHAR(50)
zone_id INT
dim_zone
zone_id INT
zone_code VARCHAR(50)
zone VARCHAR(50)
city VARCHAR(50)
dbdiagram.io

`


Home
ChatBot
AI/ML Graphs
Welcome to Pharmalytica AI Dashboard
Select a feature from the top navigation to begin.
About Pharmalytica
Pharmalytica is a cutting-edge analytics platform built to empower pharmaceutical professionals with actionable insights derived from complex medical and sales data. Our goal is to simplify decision-making using AI-driven tools, interactive dashboards, and natural language interfaces. This project combines advanced data visualization, real-time AI assistance, and intelligent data processing to support pharmacies, manufacturers, and healthcare institutions in optimizing their operations and understanding key market trends.


ChatBot
AI Assistant
Hello! I'm your AI assistant. How can I help you today?
07:39 PM
Who are the data modelers and what are they responsible for
07:39 PM
Data Modelers are Ahmed Ibrahim and Yassin Elmaghrabi. They are responsible for: * Designing the data model and ensuring the database schema matches business needs. * Defining and building the underlying data structures, entity relationships, and database schema designs that support query efficiency and data integrity. * Ensuring data models reflect real-world business concepts like product taxonomy, sales transactions, and pharmacy hierarchies. * Interfacing with the Project Initiator – Dashboard Objective Lead to ensure data pipelines and models support dashboard requirements.
07:39 PM
Type your message...

`

Data Engineers:

Phase 1: Collect data and explore:

In this phase, the two of us are searching for good, relevant data for our project idea in Kaggle. After we find it and agree with the other team members, we begin to explore it to see if there are any issues.
Then we add table insurance to our source and add columns to make the data in the future more understandable.

Phase 2: Data Preparation

- Performed **data preparation** by extracting and cleaning a random sample of data based on group key , reducing the dataset to **2.75k records** for efficient processing and analysis using Pyspark on Colab.

- Handling column names and adding columns.

- Split our one big table into 5 tables to make the ETL phase easier and to map them in Power Center and SSIS correctly

`

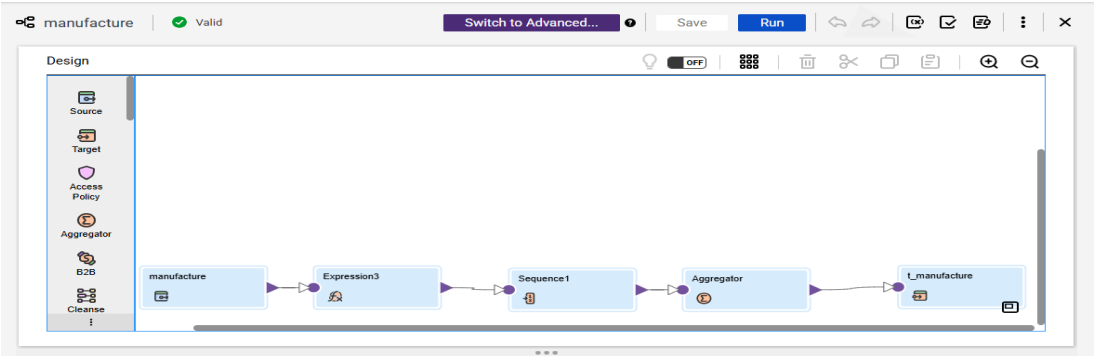
Pharmalytica

Phase 3: SSIS & Power Center (ETL):

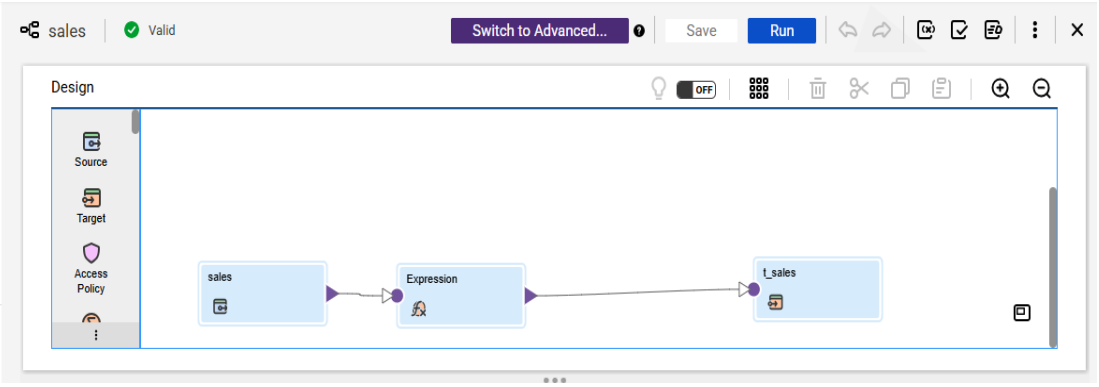
Power Center by Wafaa Ali
Landing in Power Center:

- **Purpose**: Collect raw data as-is from source systems (csv)
- **Informatica Tasks**:
  - Created **Source Connections** to extract data from operational systems.
  - Loaded data into **Landing tables** (database schema).
  - Applied **basic validations** (e.g., null checks, data type checks).
  - Ensured **truncate loading**.
- **Example**:

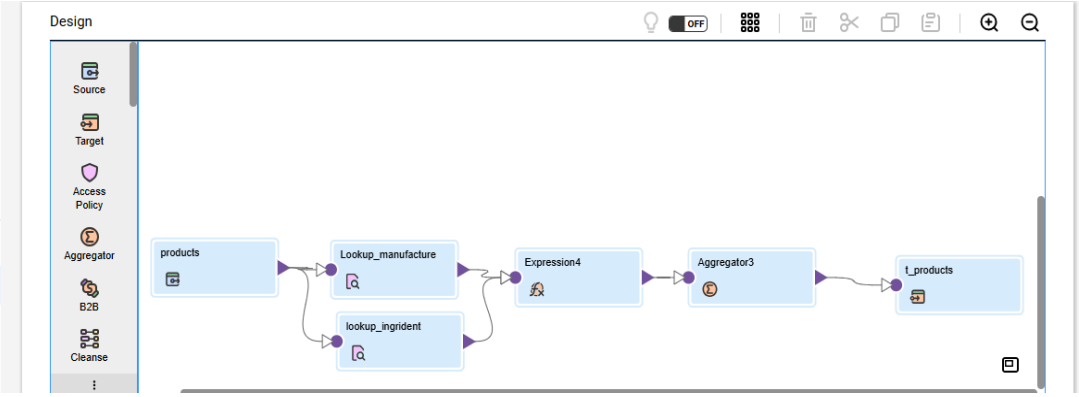
    - Landing into manufacture table



    - Landing into sales table



    - Landing into products table



`

Staging in Power Center:

- **Purpose**:
  - Cleanse, standardize, and prepare data from the Landing layer before loading into the Data Warehouse.
  - Handle **Inserts, Updates, and Deletes** using Change Data Capture (CDC).
  - Apply business rules and transformations to ensure data quality.
- **Informatica Tasks**:
  - Designed **Mappings** to move data from Landing to Staging.
  - Implemented **Lookup Transformations** on business keys to detect Inserts, Updates, and Deletes.
  - Used **Expression and Router Transformations** to classify rows (NEW, UPDATED, DELETED).
  - Applied **Update Strategy** for proper action (Insert, Update, Delete/Soft Delete).
  - Ensured **incremental loading** (capture only changes from source instead of full reload).

1. Inserts (New Records)

- **Definition**: Records that exist in Landing but not in Staging.
- **Action**: Insert new records into Staging with is_current = 1 and status = NEW.
- **Implementation**:
  - Performed **Lookup on business key** (e.g., product_id, group_key) in Staging.
  - If no match found → flag as NEW (Insert).
  - Use **Update Strategy (DD_INSERT)** to load into Staging.

2. Updates (Modified Records)

- **Definition**: Records that exist in both Landing and Staging, but attribute values have changed (e.g., price, name).
- **Action**: Insert the record in Staging with is_current = 1 and status = MODIFIED , old record updated with is_current = 0 and status is NEW.
- **Implementation**:
  - Compare key attributes between Landing and Staging.
  - If difference detected → flag as MODIFIED (Update).
  - Use **Update Strategy (DD_UPDATE)** to apply changes.

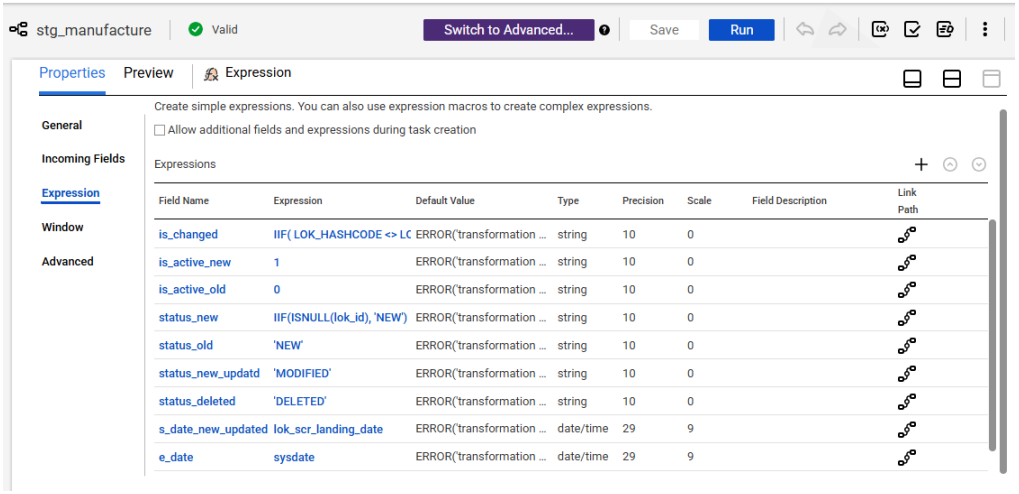
`

3. Deletes (Removed Records)

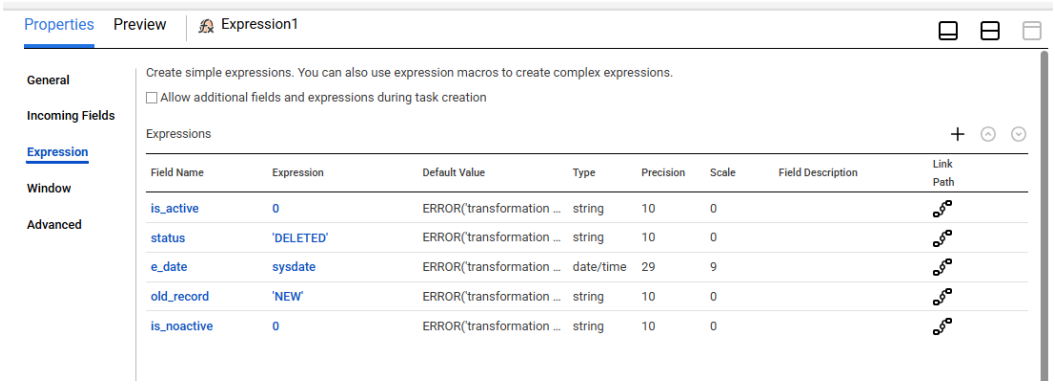
- **Definition**: Records that exist in Staging but are not present in Landing during incremental load.
- **Action**:
  - **Soft Delete (preferred)**: Mark old record as inactive (is_current = 0) and insert new record with status = DELETED.
- **Implementation**:
  - Full Outer Join between Landing and Staging.
  - Records missing in Landing flagged as (Deleted).
  - Use **Update Strategy (DD_DELETE)** for physical delete or update a flag for soft delete.

- **Example:**

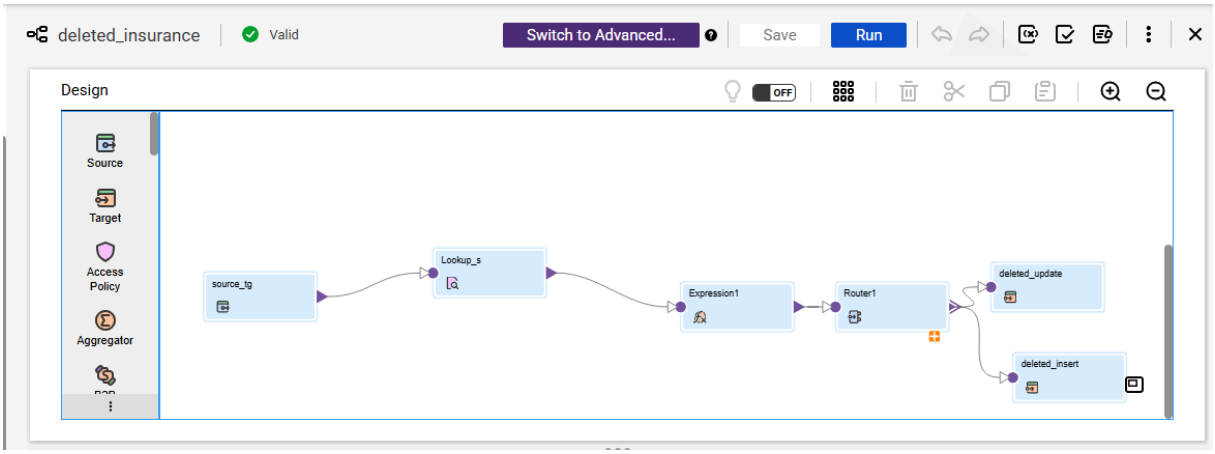
- Expression handling for insert and updated

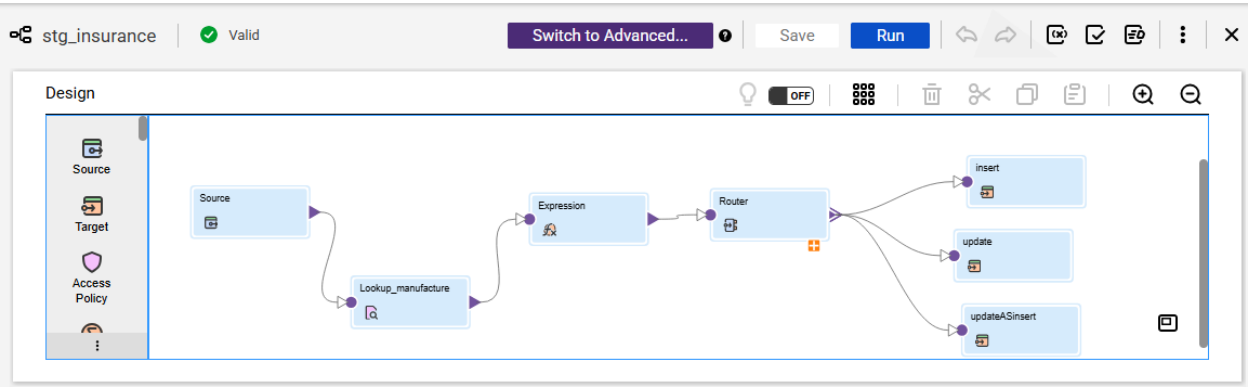


- Expression handling for deleted

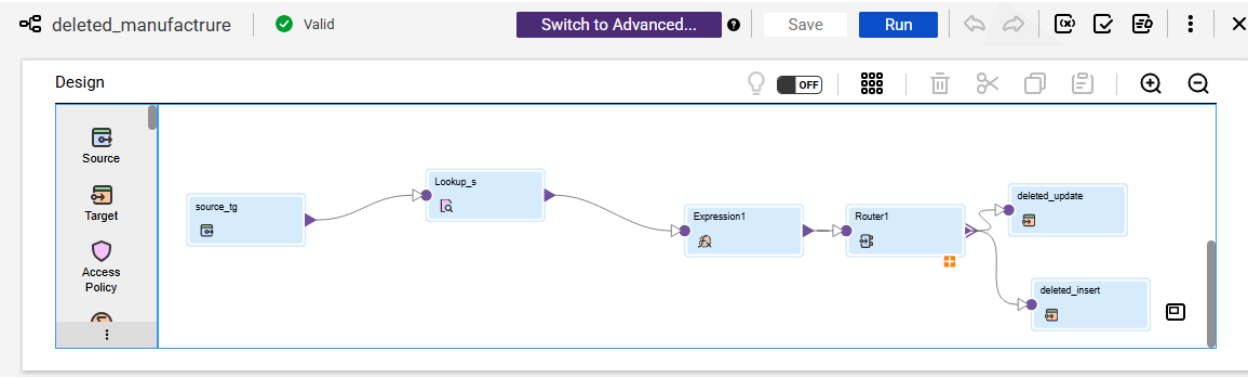


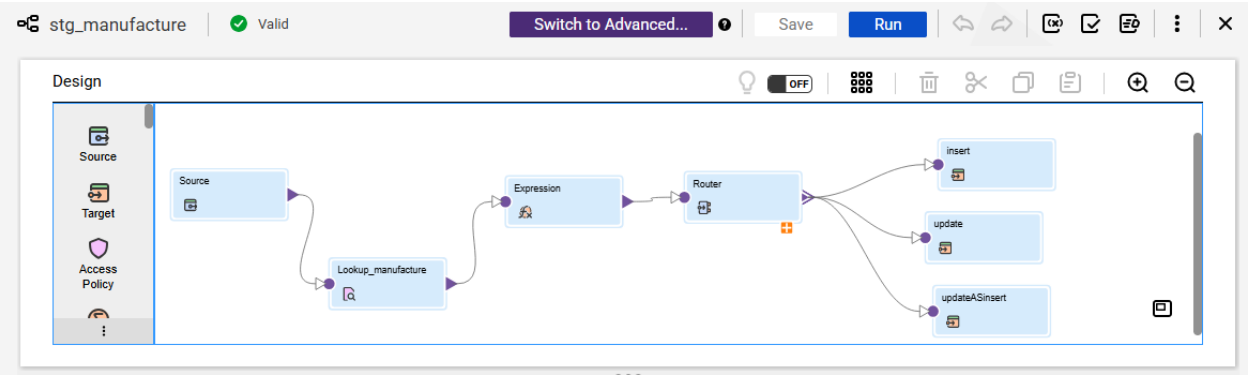
- Mapping update and delete for insurance





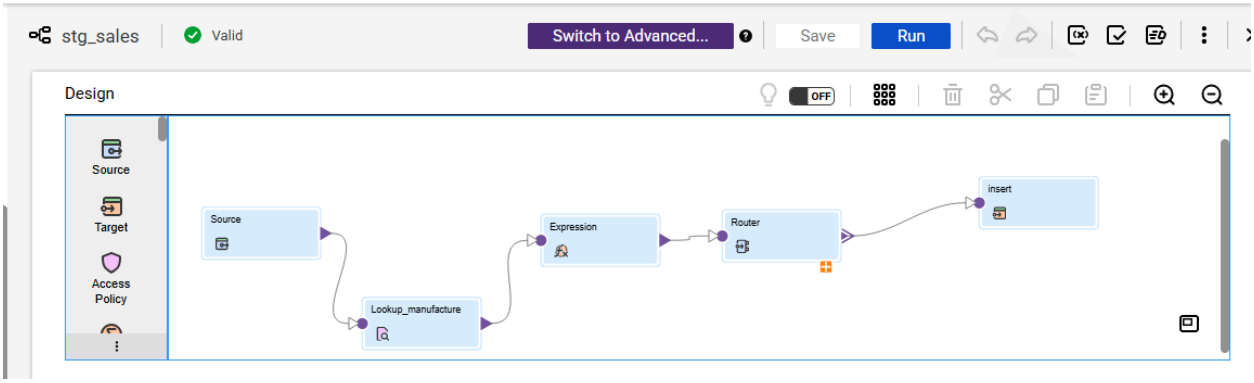
- Updated and deleted for manufacture

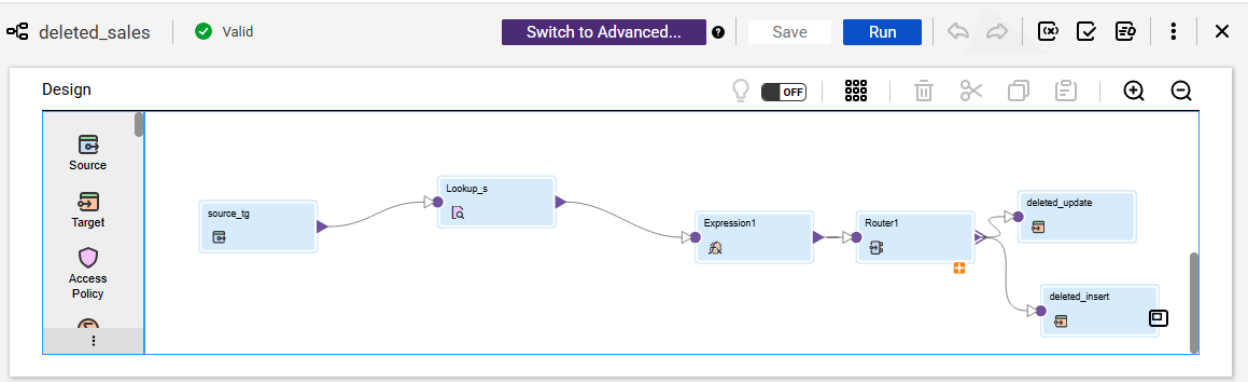




`

- Updated and deleted for sales table





- **Test:**

  - Updated

	insurance_sk	id	insurance_name	average_coverage	landing_date	start_date	end_date	is_current	record_status	hashcode
1	35	17	layan	0.15	2025-08-16	2025-08-16	NULL	1	MODIFIED	9AE78C513B

	insurance_sk	id	insurance_name	average_coverage	landing_date	start_date	end_date	is_current	record_status	hashcode
1	34	17	wafaa	0.15	2025-08-16	2025-08-16	2025-08-16	0	NEW	CBB4014949

  - Deleted

100 % 44 0

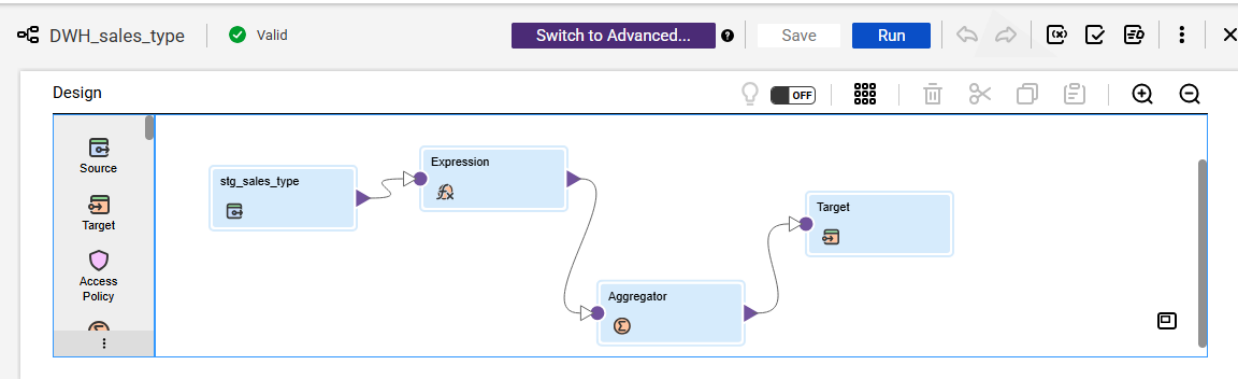
	insurance_sk	id	insurance_name	average_coverage	landing_date	start_date	end_date	is_current	record_status	hashcode
1	35	17	layan	0.15	2025-08-16	2025-08-16	2025-08-16	0	NEW	9AE78C513B
2	37	17	layan	0.15	2025-08-16	2025-08-16	2025-08-16	0	DELETED	NULL

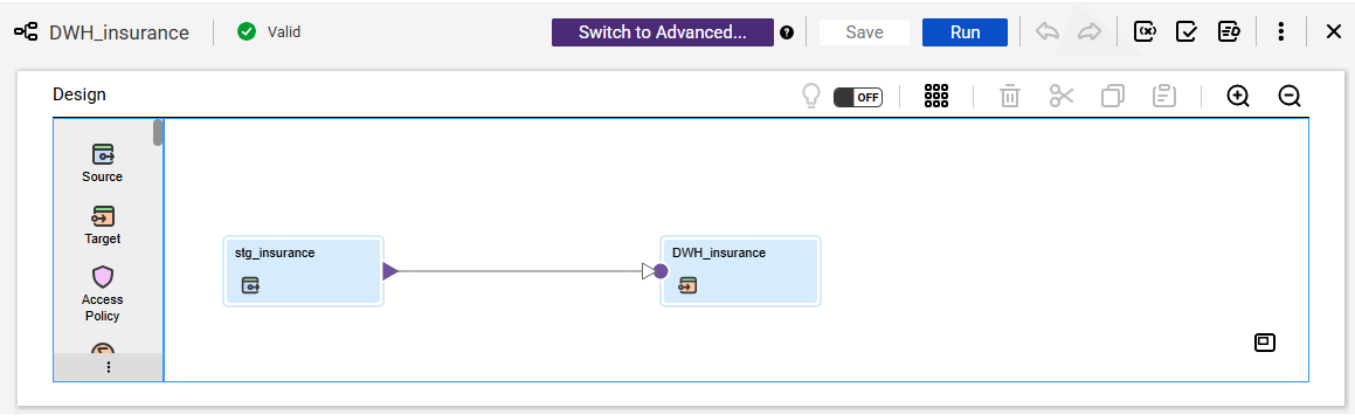
`

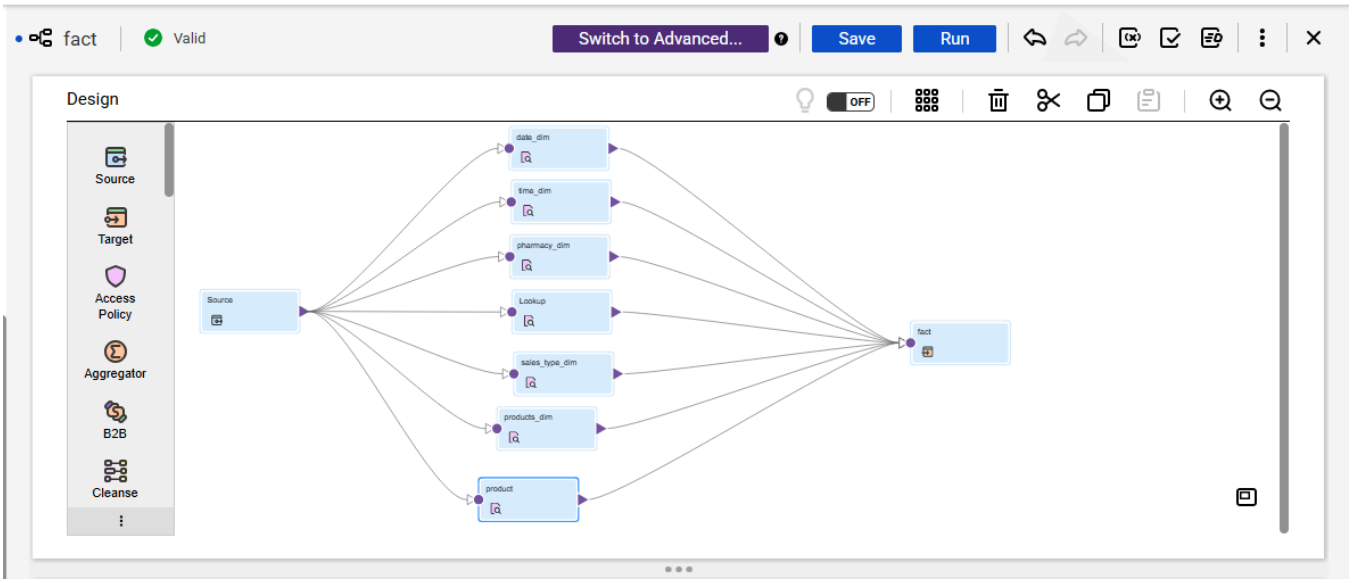
DWH in Power Center:

- **Purpose**:
  - Store integrated, cleansed, and historical data in a **dimensional model (Snowflake Schema)**.
  - Enable **reporting, KPIs, and advanced analytics**.
  - Maintain historical tracking via **Slowly Changing Dimensions (SCDs)**.
- **Informatica Tasks**:
  - Designed mappings to load **Dimension Tables**:
    - dim_product, dim_pharmacy, dim_manufacturer, dim_date, dim_sales_type.
  - Loaded **Fact Tables**:
    - fact_sales with measures (sales quantity, revenue, invoice count, etc.).
  - Managed **surrogate keys** (product_sk, pharmacy_sk, etc.) for dimensional modeling.
  - Implemented **SCD logic**:
    - **Type 1** (overwrite values).
    - **Type 2** (track history with start_date, end_date, is_current).
  - Ensured **referential integrity** (facts must always link to valid dimensions).
  - Applied **truncate and load.**

- **Example:**





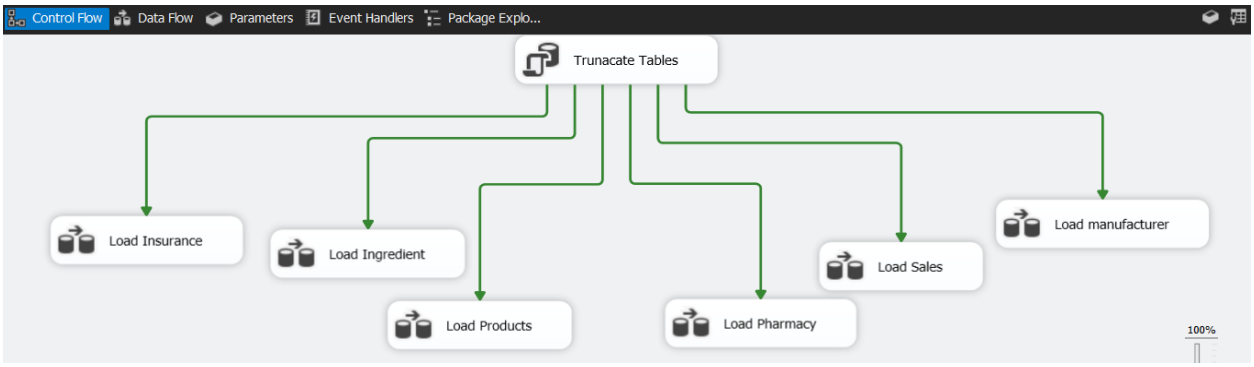


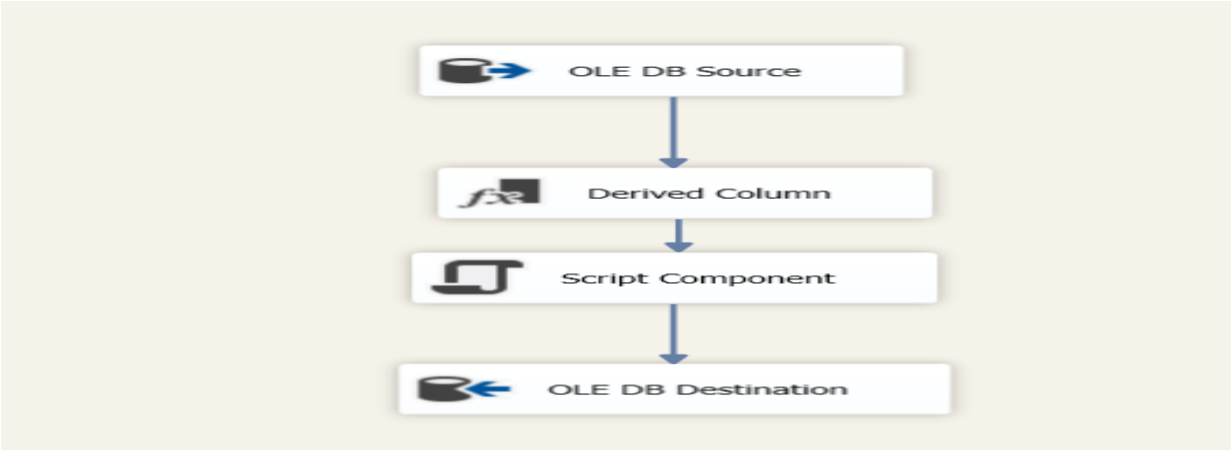
Pharmalytica

SSIS by Abdallah Maher:

Landing in SSIS:

In this layer, we use the initial load and truncate and load to transfer data from the source to the landing tables, allowing us to make changes or revert to the landing if issues arise in the next steps. Also, we added the landing date to know the last time we took the last update of data and add hash code to track the history of data in the Staging layer.

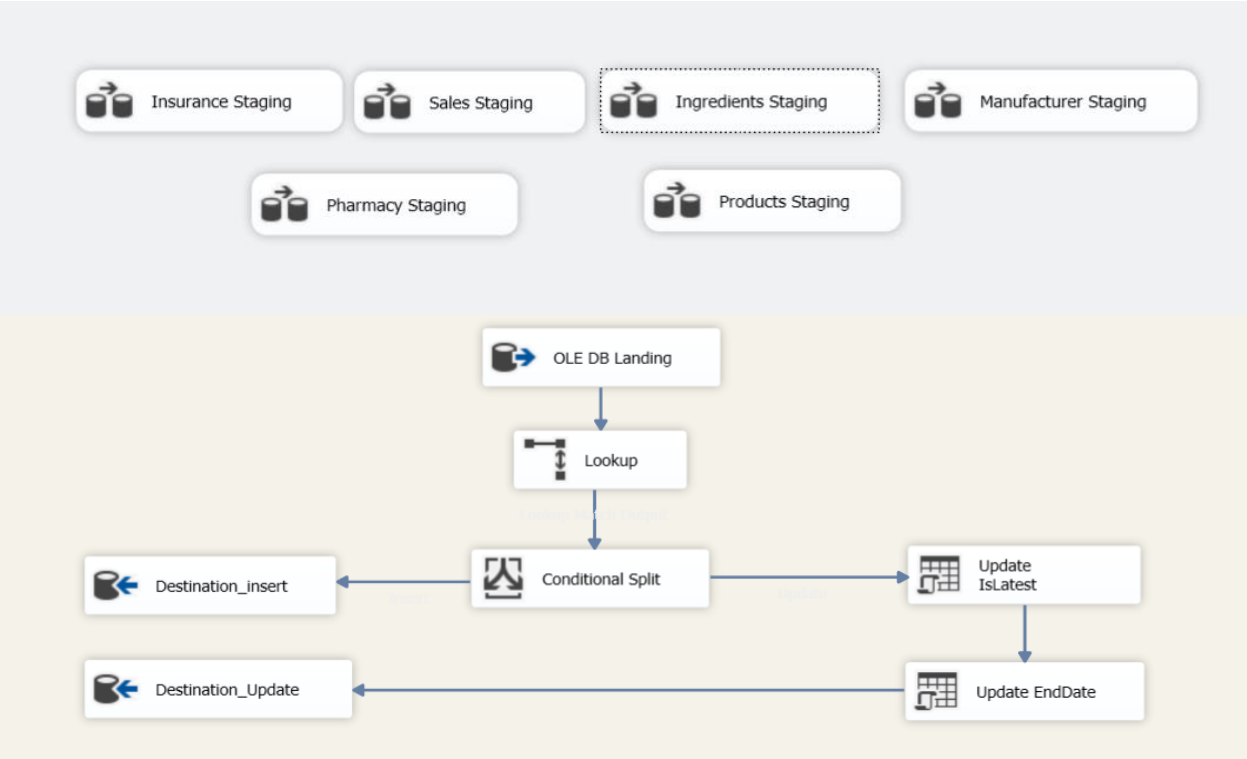




`

Staging in SSIS:

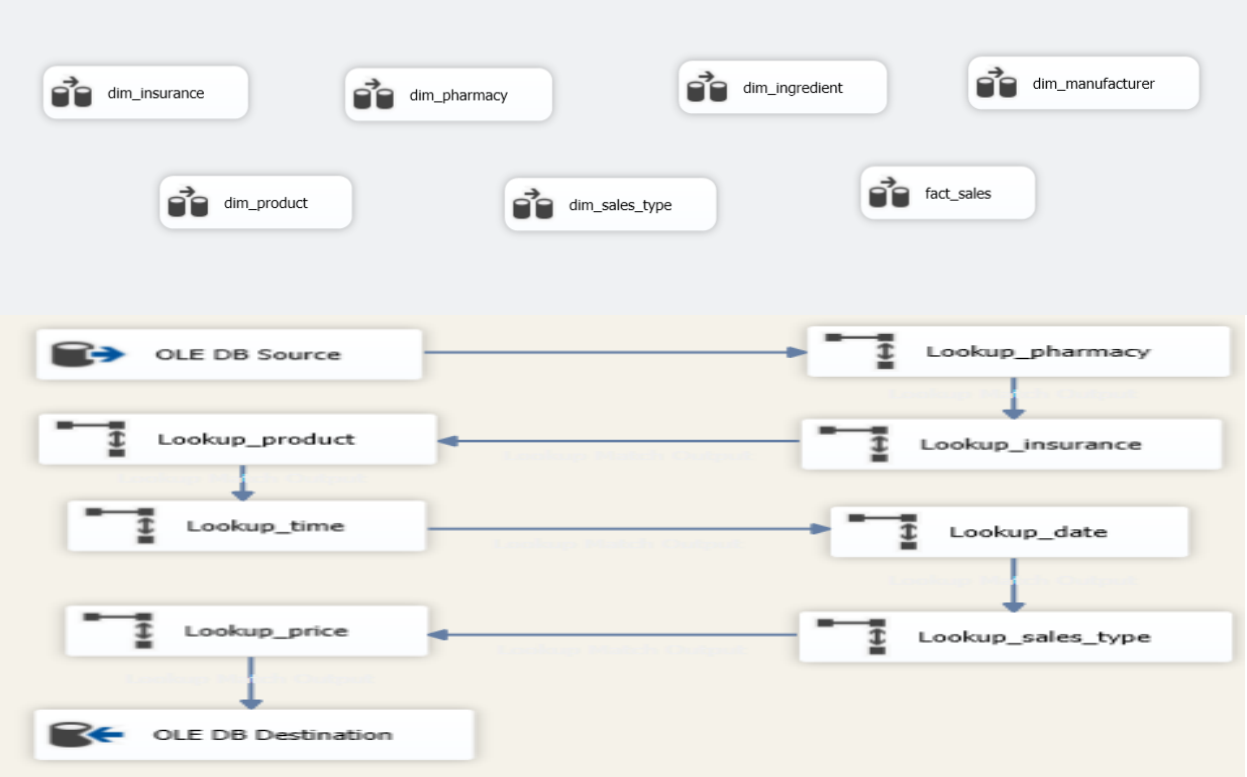
In this layer, we track the history of (UPDATE, INSERT, and DELETE) to show these changes in the tables of staging, and we use a hash code to track these changes using "conditional split" to split update rows and insert rows and show them all in the table of staging.
Also, add columns (staging date, start date, end date, is latest, and is deleted) to help understand the data history.



DWH in SSIS:

In this layer, we add SK and BK to the DWH tables and add a date and time dimension to the data, as we mentioned in the Schema.
We use "lookup" in dim_products and fact_sales to link and map these tables well with the others to get true insights and avoid data misleading.

`



Phase 4: Testing:

Measure	Landing	Query	Satging	Query	DWH	Query	Status
Total Revenue	49884013	use pharmalitica SELECT SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) AS total_value FROM sales s JOIN products p ON s.product_id = p.product_id;	49884013	use st_pharmalitica select SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) AS total_value from stg_sales_data s join stg_products_data p on s.product_id = p.product_id	49884013	use DWH SELECT CAST(SUM(f.price * (CAST(f.sales_sheet AS DECIMAL(18,2)) / NULLIF(f.sheet, 0))) AS DECIMAL(18,2)) AS total_revenue FROM fact_sales f	**ACCEPTED**
Avg rev for phamacies	1385667	use pharmalitica SELECT SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0)))/36 AS total_value_per_pharmacy FROM sales s JOIN products p ON s.product_id = p.product_id;	1385667	use st_pharmalitica select SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) /36 AS total_value_per_pharmacy from stg_sales_data s join stg_products_data p on s.product_id = p.product_id	1385667	use DWH SELECT CAST(SUM(f.price * (CAST(f.sales_sheet AS DECIMAL(18,2)) / NULLIF(f.sheet, 0))) AS DECIMAL(18,2)) /36 AS total_value_per_pharmacy FROM fact_sales f	**ACCEPTED**
Avg rev for Quarter	12471010	use pharmalitica SELECT SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0)))/4 AS total_value_per_pharmacy FROM sales s JOIN products p ON s.product_id = p.product_id;	12471010	use st_pharmalitica select SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) /4 AS total_value_per_pharmacy from stg_sales_data s join stg_products_data p on s.product_id = p.product_id	12471010	SELECT CAST(SUM(f.price * (CAST(f.sales_sheet AS DECIMAL(18,2)) / NULLIF(f.sheet, 0))) AS DECIMAL(18,2)) /4 AS total_value_per_pharmacy FROM fact_sales f	**ACCEPTED**
Avg rev per invoice	217.41922	use pharmalitica SELECT CAST(SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) AS DECIMAL(18,2)) / COUNT(DISTINCT s.invoice) as avg_per_invoice	217.41922	use st_pharmalitica SELECT CAST(SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) AS DECIMAL(18,2)) / COUNT(DISTINCT s.invoice) as avg_per_invoice FROM stg_sales_Data s JOIN .stg_products_data p	217.41922	use DWH SELECT CAST(SUM(f.price * (CAST(f.sales_sheet AS DECIMAL(18,2)) / NULLIF(f.sheet, 0))) / NULLIF(COUNT(DISTINCT f.invoice_bk), 0) AS DECIMAL(18,2)) AS avg_per_invoice	**ACCEPTED**

`

Avg sold units per day	830.774376	select sum(s.sales_sheet/s.sheet) /365 as Avg_sold_unit from sales_Landing s;	830.774376	select sum(s.sales_sheet/s.sheet) /365 as Avg_sold_unit from sales_Staging s;	830.774376	select sum(s.sales_sheet/s.sheet) /365 as Avg_sold_unit from fact_sales s;	**ACCEPTED**
Count Invoices	229437	use pharmalitica SELECT COUNT(DISTINCT s.invoice) AS count_invoices FROM sales s;	229437	use st_pharmalitica SELECT COUNT(DISTINCT s.invoice) AS count_invoices FROM stg_sales_Data s;	229437	use DWH SELECT COUNT(DISTINCT f.invoice_bk) AS count_invoices FROM fact_sales f ;	**ACCEPTED**
Sales invoices with insurance	212401	select APPROX_COUNT_DISTINCT(invoice) as Invoices_with_Insurance from sales_landing where insurance_id <> 16 ;	212401	select APPROX_COUNT_DISTINCT(invoice) as Invoices_with_Insurance from sales_Staging where insurance_id <> 16 ;	212401	select APPROX_COUNT_DISTINCT(invoice_bk) as Invoices_with_Insurance from fact_sales s join dim_insurance i on i.insurance_sk = s.insurance_sk where insurance_bk <> 16 ;	**ACCEPTED**
Max Rev per day	400685.77	USE pharmalitica; GO WITH daily_revenue AS (SELECT CAST(s.addeddate AS DATE) AS sales_day, SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet,0))) AS total_revenue FROM sales s JOIN products p ON s.product_id = p.product_id GROUP BY CAST(s.addeddate AS DATE)) SELECT MAX(total_revenue) AS max_revenue_per_day FROM daily_revenue;	400685.77	FROM daily_revenue; USE st_pharmalitica; GO WITH daily_revenue AS (SELECT CAST(s.addeddate AS DATE) AS sales_day, SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet,0))) AS total_revenue FROM stg_sales_Data s JOIN stg_products_data p ON s.product_id = p.product_id GROUP BY CAST(s.addeddate AS DATE)) SELECT MAX(total_revenue) AS max_revenue_per_day FROM daily_revenue;	400685.77	USE DWH; GO WITH daily_revenue AS (SELECT d.full_date AS sales_day, -- from dim_date SUM(f.price * (CAST(f.sales_sheet AS DECIMAL(18,2)) / NULLIF(f.sheet,0))) AS total_revenue FROM fact_sales f JOIN dim_date d ON f.date_sk = d.date_sk GROUP BY d.full_date) SELECT MAX(daily_revenue.total_revenue) AS max_revenue_per_day FROM daily_revenue;	**ACCEPTED**

Min Rev per day	66061.18204	USE pharmalitica; GO WITH daily_revenue AS (SELECT CAST(r.addeddate AS DATE) AS sales_day, SUM(p.price_inr * (CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0))) AS total_revenue FROM sales r JOIN products p ON r.product_id = p.product_id GROUP BY CAST(r.addeddate AS DATE)) SELECT MIN(total_revenue) AS min_revenue_per_day FROM daily_revenue;	66061.18204	FROM daily_revenue; USE st_pharmalitica; GO WITH daily_revenue AS (SELECT CAST(r.addeddate AS DATE) AS sales_day, SUM(p.price_inr * (CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0))) AS total_revenue FROM stg_sales_Data r JOIN stg_products_data p ON r.product_id = p.product_id GROUP BY CAST(r.addeddate AS DATE)) SELECT MIN(total_revenue) AS min_revenue_per_day FROM daily_revenue;	66061.18204	USE DWH; GO WITH daily_revenue AS (SELECT d.full_date AS sales_day, -- from dim_date SUM(f.price * (CAST(f.sales_sheet AS DECIMAL(18,2)) / NULLIF(f.sheet,0))) AS total_revenue FROM fact_sales f JOIN dim_date d ON f.date_sk = d.date_sk GROUP BY d.full_date) SELECT MIN(daily_revenue.total_revenue) AS min_revenue_per_day FROM daily_revenue;	**ACCEPTED**
Top rev manufacturer	Biacan	SELECT TOP 1 m.name, SUM(p.price_inr * (CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0))) AS total_revenue FROM sales_Landing r JOIN products_Landing p ON r.product_id = p.product_id JOIN manufacturer_Landing m ON p.manufacturer_id = m.id GROUP BY m.name ORDER BY total_revenue DESC;	Biacan	SELECT TOP 1 m.name, SUM(p.price_inr * (CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0))) AS total_revenue FROM sales_Staging r JOIN products_Staging p ON r.product_id = p.product_id JOIN manufacturer_staging m ON p.manufacturer_id = m.id GROUP BY m.name ORDER BY total_revenue DESC;	Biacan	SELECT TOP 1 m.manufacturer_name, SUM(r.Price * (CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0))) AS total_revenue FROM fact_sales r JOIN dim_product p ON r.product_rk = p.product_rk JOIN dim_manufacturer m ON p.manufacturer_rk = m.manufacturer_rk GROUP BY m.manufacturer_name ORDER BY total_revenue DESC;	**ACCEPTED**
Top rev City	03	USE pharmalitica; GO WITH city_revenue AS (SELECT ph.city_id, SUM(p.price_inr * (CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0))) AS total_revenue FROM sales r JOIN products p ON r.product_id = p.product_id JOIN pharmacy ph ON r.group_key = ph.group_key GROUP BY ph.city_id) SELECT TOP 1 city_id, total_revenue FROM city_revenue ORDER BY total_revenue DESC;	03	USE st_pharmalitica; GO WITH city_revenue AS (SELECT ph.city_id, SUM(p.price_inr * (CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0))) AS total_revenue FROM stg_sales_Data r JOIN stg_products_data p ON r.product_id = p.product_id JOIN stg_pharmacy_Data ph ON r.group_key = ph.group_key GROUP BY ph.city_id) SELECT TOP 1 city_id, total_revenue FROM city_revenue ORDER BY total_revenue DESC;	Giza	USE DWH; GO WITH city_revenue AS (SELECT ph.city, SUM(f.price * (CAST(f.sales_sheet AS DECIMAL(18,2)) / NULLIF(f.sheet,0))) AS total_revenue FROM fact_sales f JOIN dim_pharmacy ph ON f.pharmacy_rk = ph.pharmacy_rk GROUP BY ph.city) SELECT TOP 1 city, total_revenue FROM city_revenue ORDER BY total_revenue DESC;	**ACCEPTED**
Total Sold Units	303232.6472	SELECT SUM(CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0)) AS total_sold_units FROM sales_landing r;	303232.6472	SELECT SUM(CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0)) AS total_sold_units FROM sales_staging r;	303232.6472	SELECT SUM(CAST(r.sales_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet,0)) AS total_sold_units FROM fact_sales r;	**ACCEPTED**

`

Top Selling Product	am c	SELECT TOP 10 p.clean_brand_name, SUM(CAST(r.relase_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet, 0)) AS total_relase_units FROM relase_landing r JOIN products_landing p ON p.product_id = r.product_id GROUP BY p.clean_brand_name ORDER BY total_relase_units DESC;	am c	SELECT TOP 10 p.clean_brand_name, SUM(CAST(r.relase_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet, 0)) AS total_relase_units FROM relase_Staging r JOIN products_Staging p ON p.product_id = r.product_id GROUP BY p.clean_brand_name ORDER BY total_relase_units DESC;	am c	SELECT TOP 10 p.clean_brand_name, SUM(CAST(r.relase_sheet AS DECIMAL(18,2)) / NULLIF(r.sheet, 0)) AS total_relase_units FROM fact_relase r JOIN dim_product p ON p.product_sk = r.product_sk GROUP BY p.clean_brand_name ORDER BY total_relase_units DESC;	ACCEPTED
Total_Cities	3	USE pharmalitica; SELECT COUNT(DISTINCT city_id) AS total_cities FROM pharmacy;	3	USE st_pharmalitica; SELECT COUNT(DISTINCT city_id) AS total_cities FROM stg_pharmacy_data;	3	USE DWH SELECT COUNT(DISTINCT city) AS total_cities FROM dim_pharmacy;	ACCEPTED
Total_delivery	68781	USE pharmalitica; SELECT COUNT(DISTINCT r.invoice) AS delivery_invoices FROM relase r WHERE r.relase_type = 'delivery';	68781	USE st_pharmalitica; SELECT COUNT(DISTINCT r.invoice) AS delivery_invoices FROM stg_relase_Data r WHERE r.relase_type = 'delivery';	68781	USE DWH; SELECT COUNT(DISTINCT f.invoice_bk) AS delivery_invoices FROM fact_relase f JOIN dim_relase_type t ON f.relase_type_sk = t.relase_type_sk WHERE t.relase_type = 'delivery';	ACCEPTED
Total_ingredients	826	USE pharmalitica; SELECT COUNT(DISTINCT ingredient_id) AS total_ingredient FROM ingredients;	826	USE st_pharmalitica; SELECT COUNT(DISTINCT ingredient_id) AS total_ingredient FROM stg_ingredients_data;	826	USE DWH; SELECT COUNT(DISTINCT ingredient_sk) AS total_ingredient FROM dim_ingredient;	ACCEPTED
Total_manifactures	3279	USE pharmalitica SELECT COUNT(DISTINCT name) AS total_manufacturers FROM manufacturer m;	3279	USE st_pharmalitica SELECT COUNT(DISTINCT name) AS total_manufacturers FROM stg_manufacturer_Data m;	3279	USE DWH; SELECT COUNT(DISTINCT m.manufacturer_sk) AS total_manufacturers FROM dim_manufacturer m;	ACCEPTED
total_on_site	160656	USE pharmalitica; SELECT COUNT(DISTINCT r.invoice) AS delivery_invoices FROM relase r WHERE r.relase_type = 'on_site';	160656	USE st_pharmalitica; SELECT COUNT(DISTINCT r.invoice) AS delivery_invoices FROM stg_relase_Data r WHERE r.relase_type = 'on_site'	160656	USE DWH; SELECT COUNT(DISTINCT f.invoice_bk) AS delivery_invoices FROM fact_relase f JOIN dim_relase_type t ON f.relase_type_sk = t.relase_type_sk WHERE t.relase_type = 'on_site';	ACCEPTED
Total_Pharmacies	36	USE pharmalitica SELECT COUNT(DISTINCT group_key) AS total_pharmacy FROM pharmacy m;	36	USE st_pharmalitica SELECT COUNT(DISTINCT group_key) AS total_pharmacy FROM stg_pharmacy_Data m;	36	USE DWH SELECT COUNT(DISTINCT pharmacy_sk) AS total_pharmacy FROM dim_pharmacy m;	ACCEPTED
Total_Products	11669	USE pharmalitica; SELECT COUNT(DISTINCT product_id) AS total_product FROM products;	11669	USE st_pharmalitica; SELECT COUNT(DISTINCT product_id) AS total_product FROM stg_products_data;	11669	USE DWH; SELECT COUNT(DISTINCT product_sk) AS total_product FROM dim_product;	ACCEPTED
Daily Sales	Friday	SELECT TOP 1 DATENAME(WEEKDAY, s.addeddate) AS day_of_week, SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) AS total_sales FROM sales_Landing s JOIN products_Landing p ON s.product_id = p.product_id GROUP BY DATENAME(WEEKDAY, s.addeddate) ORDER BY total_sales DESC;	Friday	SELECT TOP 1 DATENAME(WEEKDAY, s.addeddate) AS day_of_week, SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) AS total_sales FROM sales_Staging s JOIN products_Staging p ON s.product_id = p.product_id GROUP BY DATENAME(WEEKDAY, s.addeddate) ORDER BY total_sales DESC;	Friday	SELECT TOP 1 DATENAME(WEEKDAY, d.date_bk) AS day_of_week, SUM(s.price * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet, 0))) AS total_sales FROM fact_sales s JOIN dim_date d ON s.date_sk = d.date_sk GROUP BY DATENAME(WEEKDAY,d.date_bk) ORDER BY total_sales DESC;	ACCEPTED
Top Sales Hours	16	use pharmalitica GO WITH hourly_revenue AS (SELECT DATEPART(HOUR, s.time_) AS sales_hour, SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet,0))) AS total_revenue FROM sales s JOIN products p ON s.product_id = p.product_id GROUP BY DATEPART(HOUR, s.time_)) SELECT TOP 1 sales_hour, total_revenue FROM hourly_revenue ORDER BY total_revenue DESC;	16	use st_pharmalitica GO WITH hourly_revenue AS (SELECT DATEPART(HOUR, s.time_) AS sales_hour, SUM(p.price_inr * (CAST(s.sales_sheet AS DECIMAL(18,2)) / NULLIF(s.sheet,0))) AS total_revenue FROM stg_sales_Data s JOIN stg_products_data p ON s.product_id = p.product_id GROUP BY DATEPART(HOUR, s.time_)) SELECT TOP 1 sales_hour, total_revenue FROM hourly_revenue ORDER BY total_revenue DESC;	16	USE DWH; GO WITH hourly_revenue AS (SELECT t.hour AS sales_hour, SUM(f.price * (CAST(f.sales_sheet AS DECIMAL(18,2)) / NULLIF(f.sheet,0))) AS total_revenue FROM fact_sales f JOIN dim_time t ON f.time_sk = t.time_sk GROUP BY t.hour) SELECT TOP 1 sales_hour, total_revenue FROM hourly_revenue ORDER BY total_revenue DESC;	ACCEPTED

`

Semantic:

Calculated Measures:

Name	DAX Formula	Explanation
Average Revenue per Day	DIVIDE([total_revenue], COUNTROWS(SUMMARIZE('date','date'[date_bk])))	Calculates the average daily revenue by dividing total revenue by the number of unique days.
avg rev for pharmacies	[total_revenue]/36	Calculates average revenue assuming there are 36 pharmacies.
avg sales per quarter	[total_revenue]/4	Calculates average quarterly revenue by dividing total revenue by 4.
avg sold units per day	SUM(sales[sold_units])/365	Calculates the average number of sold units per day over a 365-day period.
avg_rev	AVERAGEX(ALL('date'[day_of_year]),[total_revenue])	Returns the average revenue across all days ignoring filters.
avg_rev_per_invoices	[total_revenue]/[count_invoices]	Calculates the average revenue per invoice.
avg_sales_per_invoices	[Total Sold Units]/[count_invoices]	Calculates the average sold units per invoice.
count_invoices	DISTINCTCOUNT(sales[invoice_bk])	Counts the total number of unique invoices.
max_rev	MAXX(ALL('date'[day_of_year]),[total_revenue])	Returns the highest revenue value across all days ignoring filters.
min_rev	MINX(ALL('date'[day_of_year]),[total_revenue])	Returns the lowest revenue value across all days ignoring filters.
Sales Invoices With Insurance	CALCULATE(DISTINCTCOUNT(sales[invoice_bk]), FILTER(sales, RELATED(insurance[insurance_bk]) <> 16))	Counts invoices where the insurance code is not equal to 16.
Top Revenue City	VAR CityTable = SUMMARIZE('pharmacy','pharmacy'[city],'CityRevenue',[total_revenue]) VAR TopCity = TOPN(1, CityTable, [CityRevenue], DESC) RETURN MAXX(TopCity, 'pharmacy'[city])	Returns the city with the highest total revenue.
Top Revenue Day	VAR DayTable = SUMMARIZE(Sales,'date'[day_name],'RevenuePerDay',[total_revenue]) VAR TopDay = TOPN(1, DayTable, [RevenuePerDay], DESC) RETURN MAXX(TopDay, 'date'[day_name])	Returns the day of the week with the highest revenue.
Top Revenue Manufacturers	VAR ManifacturesTable = SUMMARIZE('manufacturer','manufacturer'[manufacturer_name],'CityRevenue',[total_revenue]) VAR TopManifacturer = TOPN(1, ManifacturesTable, [CityRevenue], DESC) RETURN MAXX(TopManifacturer, 'manufacturer'[manufacturer_name])	Returns the manufacturer with the highest total revenue.
Top Sales Hour	VAR HourTable = SUMMARIZE('time','time'[hour],'RevenuePerHour',[total_revenue]) VAR TopHour = TOPN(1, HourTable, [RevenuePerHour], DESC) RETURN MAXX(TopHour, 'time'[hour])	Returns the hour with the highest total revenue.
Top Selling Product	VAR ProductTable = SUMMARIZE('product','product'[brand_name],'RevenuePerProduct',[Total Sold Units]) VAR TopProduct = TOPN(1, ProductTable, [RevenuePerProduct], DESC) RETURN MAXX(TopProduct, 'product'[brand_name])	Returns the best-selling product based on sold units.
Total Sold Units	SUM(sales[sold_units])	Calculates the total number of sold units.
Total_Cities	DISTINCTCOUNT(pharmacy[city])	Counts the total number of unique cities.
total_delivery	CALCULATE(DISTINCTCOUNT(sales[invoice_bk]), FILTER(sales, RELATED(sales_type[sales_type]) = "delivery"))	Counts the total number of delivery invoices.
Total_ingredients	DISTINCTCOUNT(ingredient[ingredient_name])	Counts the number of unique ingredients.
Total_manufacturers	DISTINCTCOUNT(manufacturer[manufactur	Counts the number of unique manufacturers.

`

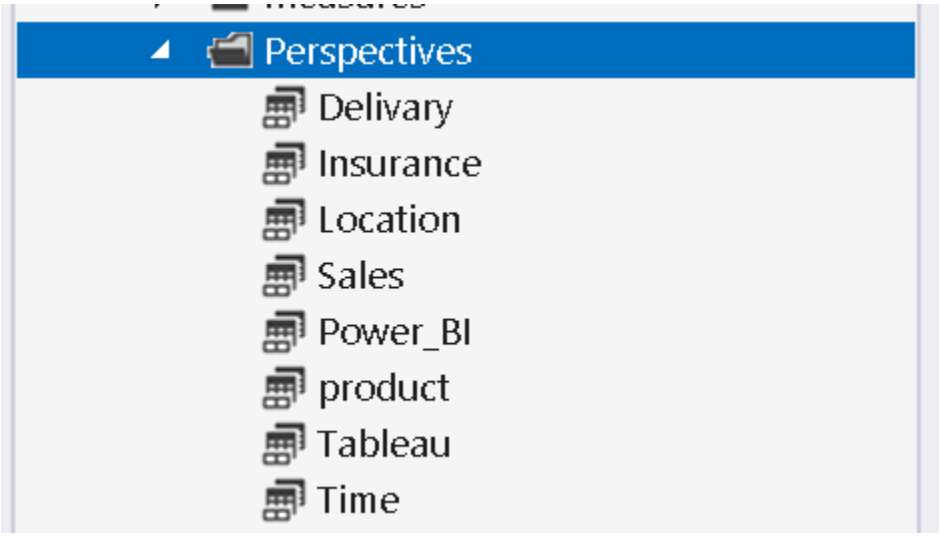
	er_name])	
total_on_site	CALCULATE(DISTINCTCOUNT(sales[invoice_bk]), FILTER(sales, RELATED(sales_type[sales_type]) = "on_site"))	Counts the total number of on-site invoices.
Total_Pharmacies	DISTINCTCOUNT(pharmacy[pharmacy_bk])	Counts the total number of unique pharmacies.
Total_Products	DISTINCTCOUNT('product'[product_sk])	Counts the number of unique products.
total_revenue	SUM(sales[Revenue])	Calculates the total revenue.
count_of_insurance_companies	CALCULATE(DISTINCTCOUNT(insurance[insurance_name]), insurance[insurance_name] <> "No Insurance")	Counts the number of insurance companies except 'No Insurance'.
total_revenue_per_delivery	CALCULATE([total_revenue],KEEPFILTERS(sales_type[sales_type] = "delivery"))	Calculates total revenue from delivery invoices only.
total_revenue_per_on_site	CALCULATE([total_revenue],KEEPFILTERS(sales_type[sales_type] = "on_site"))	Calculates total revenue from on-site invoices only.
total_sales_per_delivery	CALCULATE([Total Sold Units],KEEPFILTERS(sales_type[sales_type] = "delivery"))	Calculates total sold units for delivery invoices.
total_sales_per_on_site	CALCULATE([Total Sold Units],KEEPFILTERS(sales_type[sales_type] = "on_site"))	Calculates total sold units for on-site invoices.
Date_Converted	DATE(VALUE(LEFT('date'[date_sk],4)), VALUE(MID('date'[date_sk],5,2)), VALUE(RIGHT('date'[date_sk],2)))	Converts the date key from integer format into a proper date format.
CF Last Month	VAR _CM=[Total Sold Units] VAR _PM=CALCULATE([Total Sold Units],DATEADD('date'[Date_Converted], -1, MONTH)) VAR _perc=DIVIDE(_CM-_PM,_PM) VAR _format=SWITCH(TRUE(),_perc>0,"Green",_perc<0,"Red","Grey") RETURN _format	Returns a color indicator based on last month's performance: Green=Increase, Red=Decrease, Grey=No Change.
VS Last Month	VAR _CM=[Total Sold Units] VAR _PM=CALCULATE([Total Sold Units],DATEADD('date'[Date_Converted], -1, MONTH)) VAR _perc=DIVIDE(_CM-_PM,_PM) RETURN SWITCH(TRUE(),_perc>0,UNICHAR(9650)&" "&FORMAT(_perc,"0.0%"),_perc<0,UNICHAR(9660)&" "&FORMAT(_perc,"0.0%"),FORMAT(_perc,"0.0%"))	Displays a ▲ or ▼ arrow with percentage difference compared to last month.

Calculated Columns:

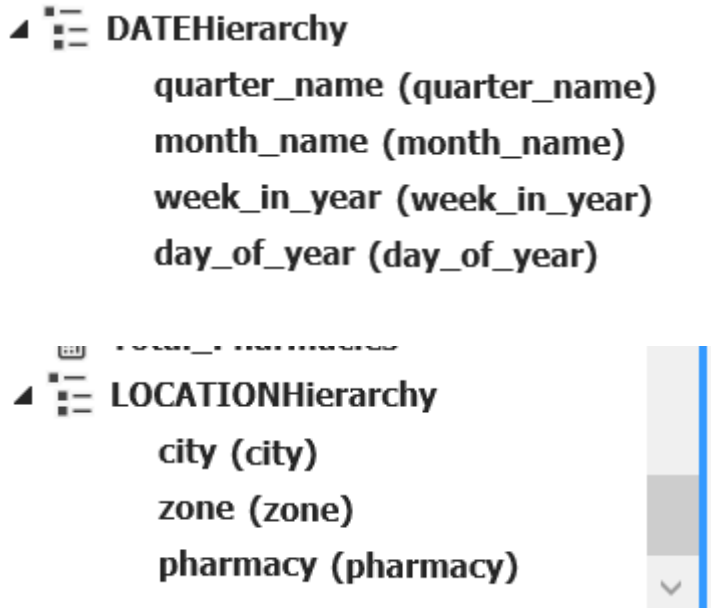
Name	DAX Formula	Explanation
Sold_units	sales[sales_sheet]/sales[sheet]	Calculates sold units as the ratio between sales_sheet and sheet columns.
Revenue	sales[sold_units]*sales[price]	Calculates revenue by multiplying sold units with price.
Day type	IF('date'[day_name] IN {"Friday","Saturday"},"Weekend","Weekday")	Classifies each day as either 'Weekend' or 'Weekday'.
super name	[city] & " - " & [zone] & " - " & [pharmacy]	Concatenates city, zone, and pharmacy name into one descriptive column.

`

Perspectives:



Hierarchies:

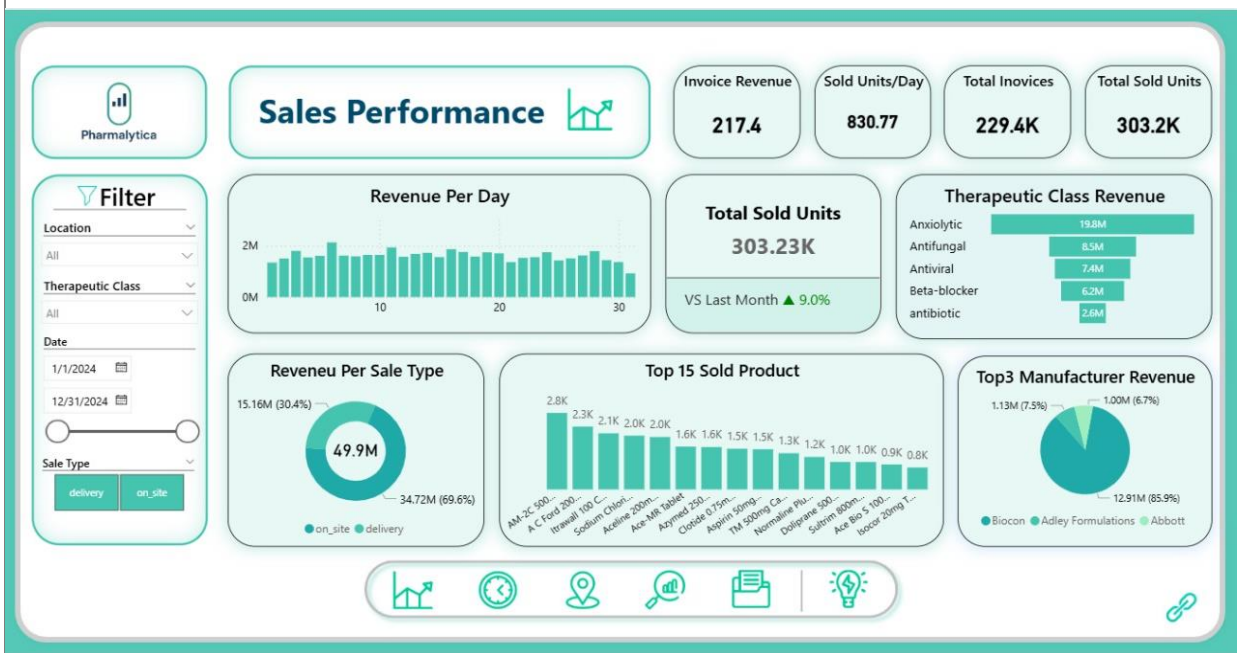


`

Data analyst:

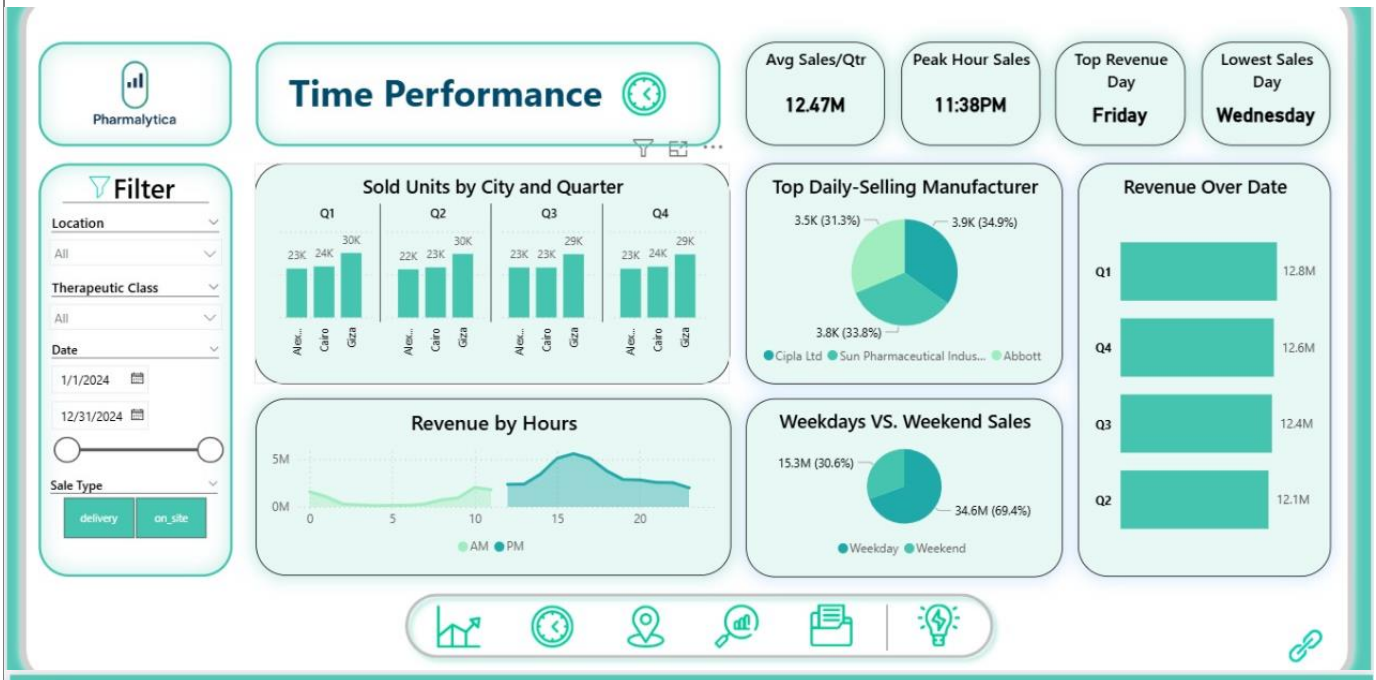
Dashboards

Sales Performance



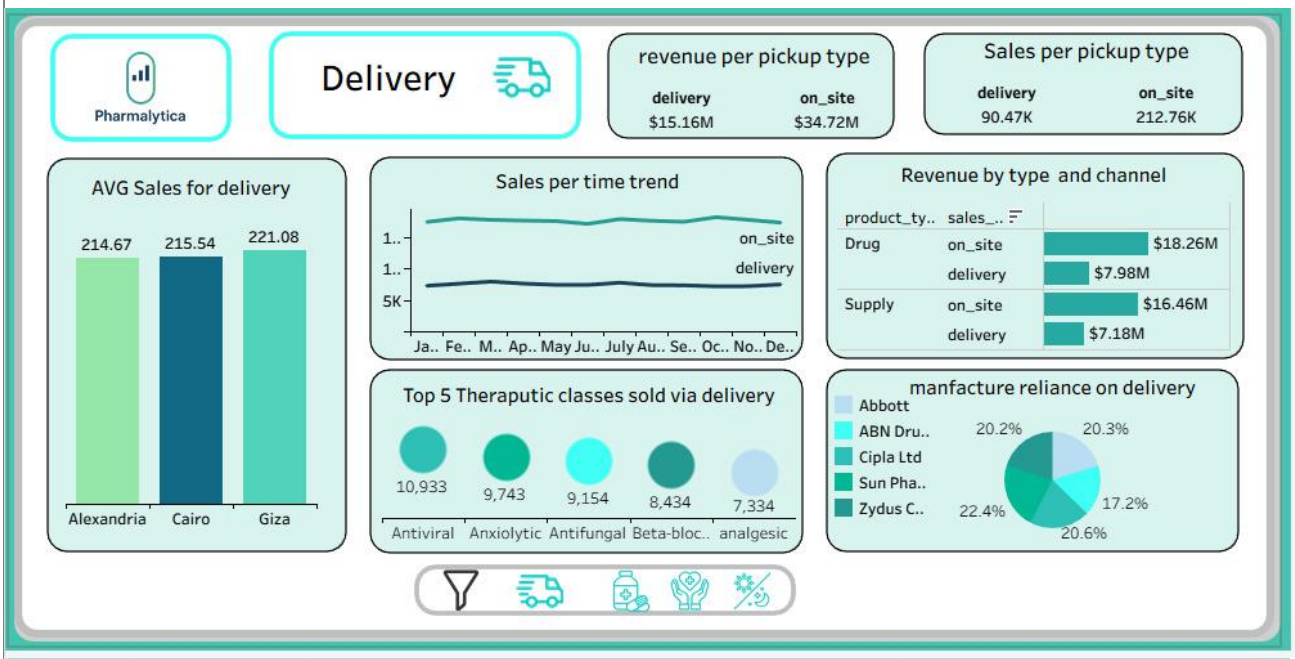
Location Performance

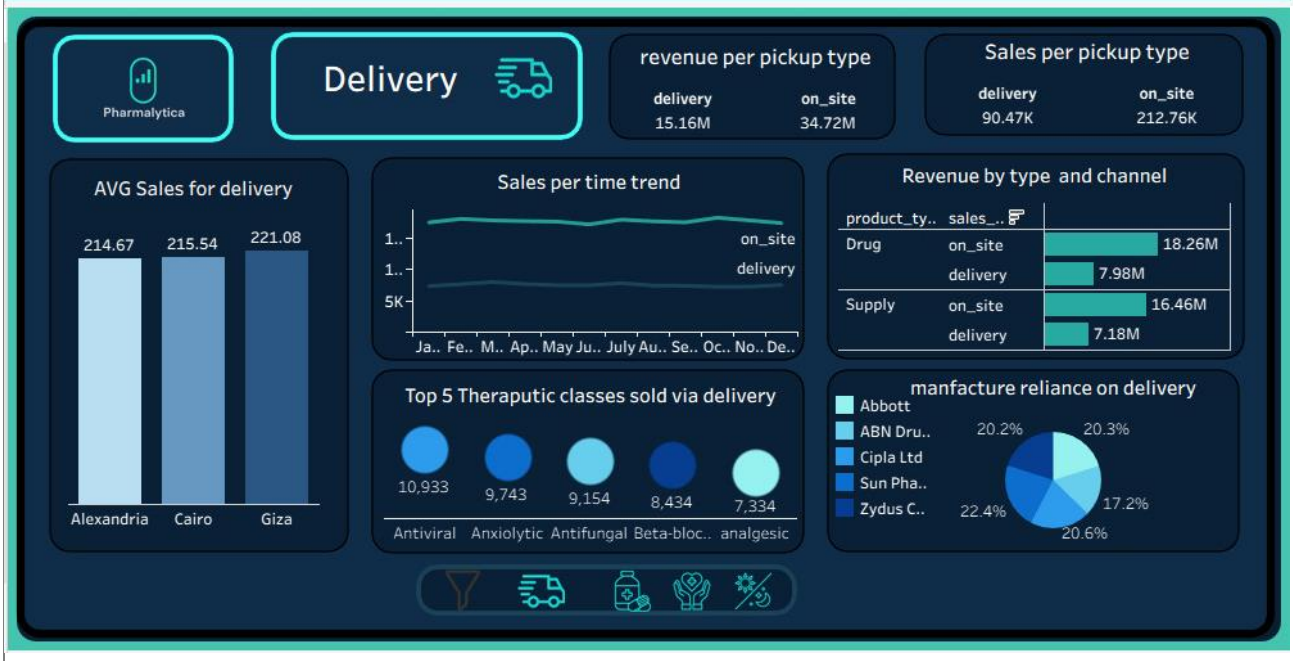
Time Performance



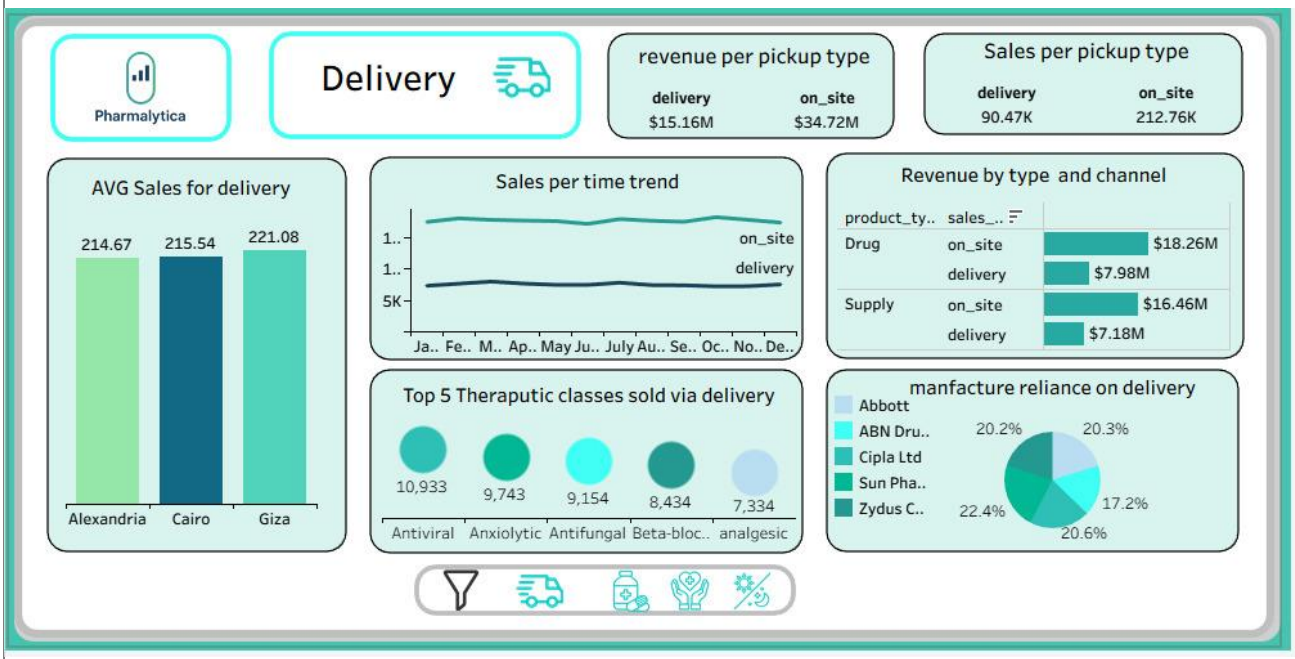
Product Performance

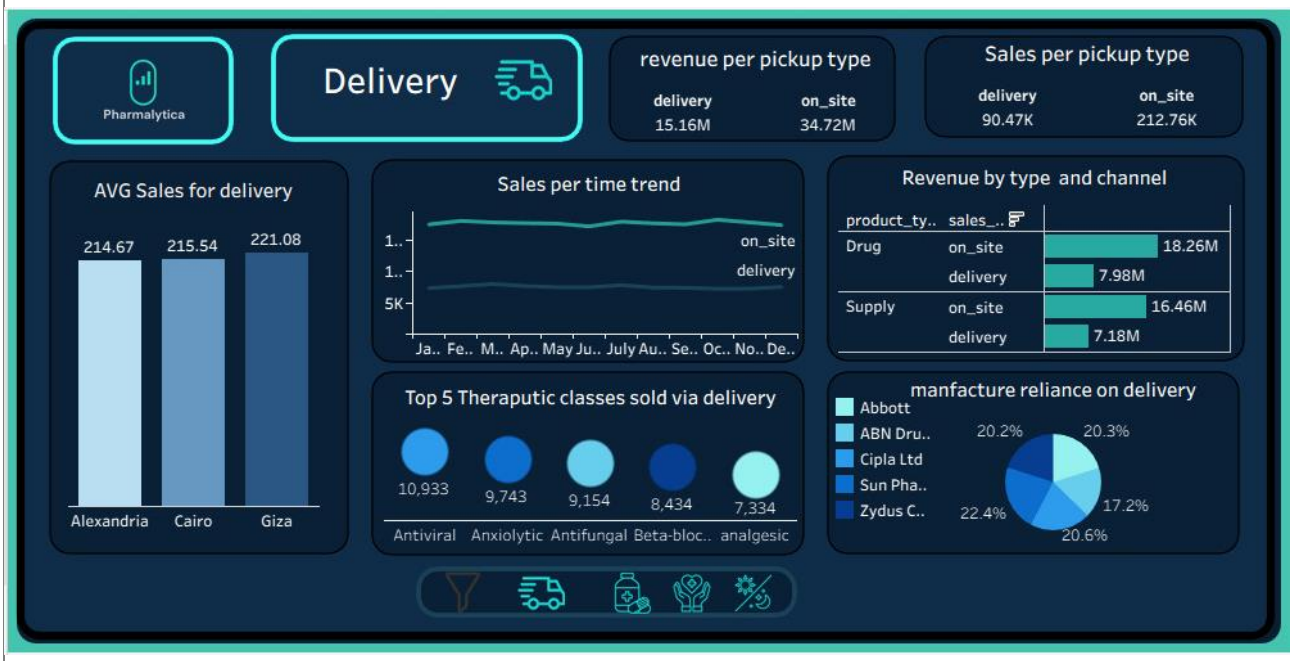
Delivery Performance



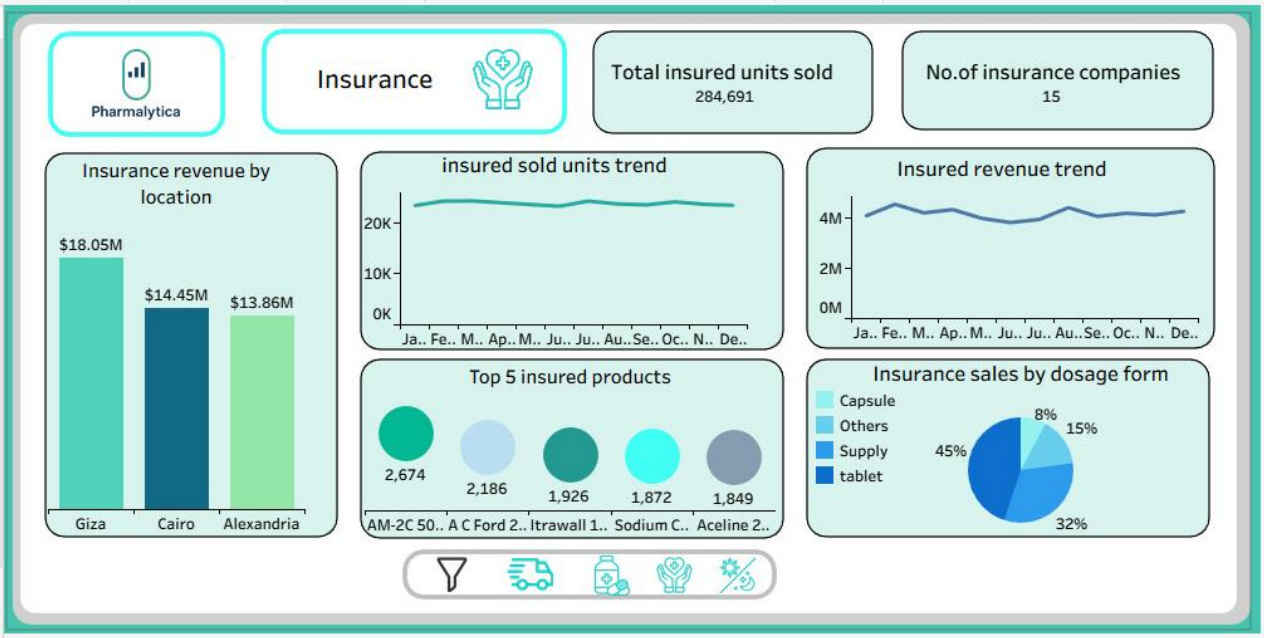


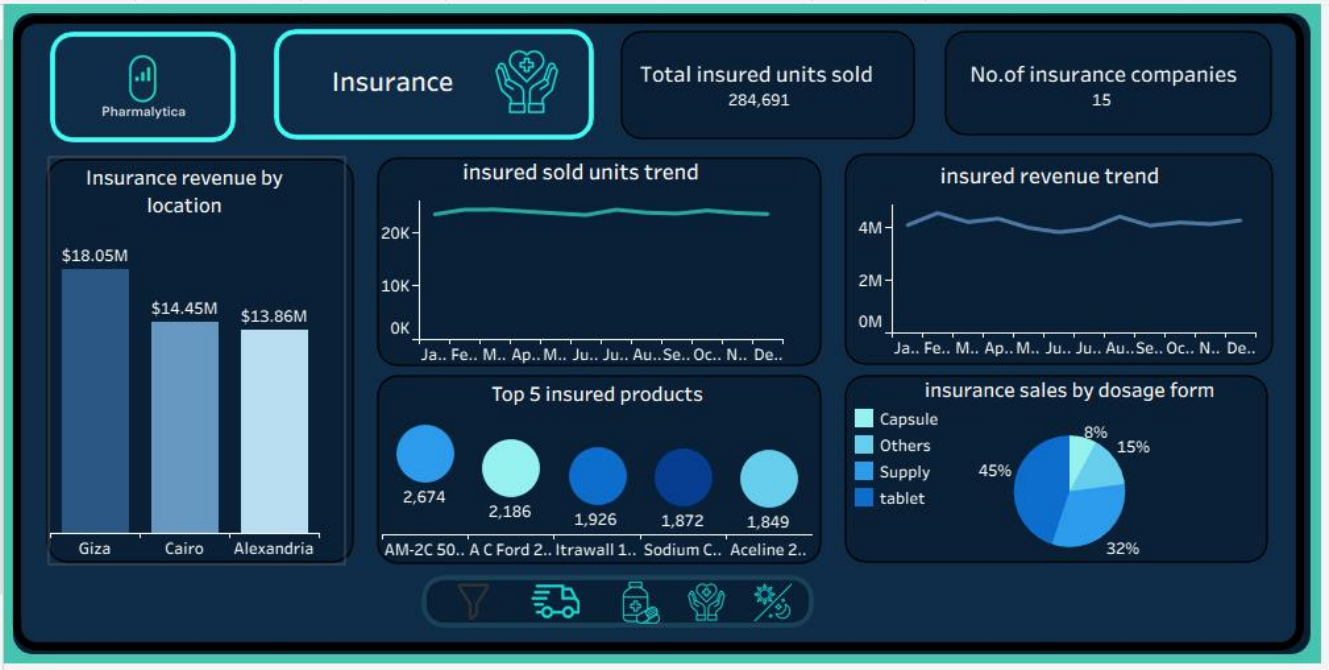
Delivery Performance


Pharmalytica
Delivery
revenue per pickup type
delivery $15.16M
on_site $34.72M
Sales per pickup type
delivery 90.47K
on_site 212.76K
AVG Sales for delivery
214.67
215.54
221.08
Alexandria
Cairo
Giza
Sales per time trend
on_site
delivery
Ja.. Fe.. M.. Ap.. May Ju.. July Au.. Se.. Oc.. No.. De..
Revenue by type and channel
product_ty.. sales_..
Drug on_site $18.26M
delivery $7.98M
Supply on_site $16.46M
delivery $7.18M
Top 5 Theraputic classes sold via delivery
10,933
9,743
9,154
8,434
7,334
Antiviral
Anxiolytic
Antifungal
Beta-bloc..
analgesic
manfacture reliance on delivery
Abbott
ABN Dru..
Cipla Ltd
Sun Pha..
Zydus C..
20.2%
20.3%
17.2%
20.6%
22.4%

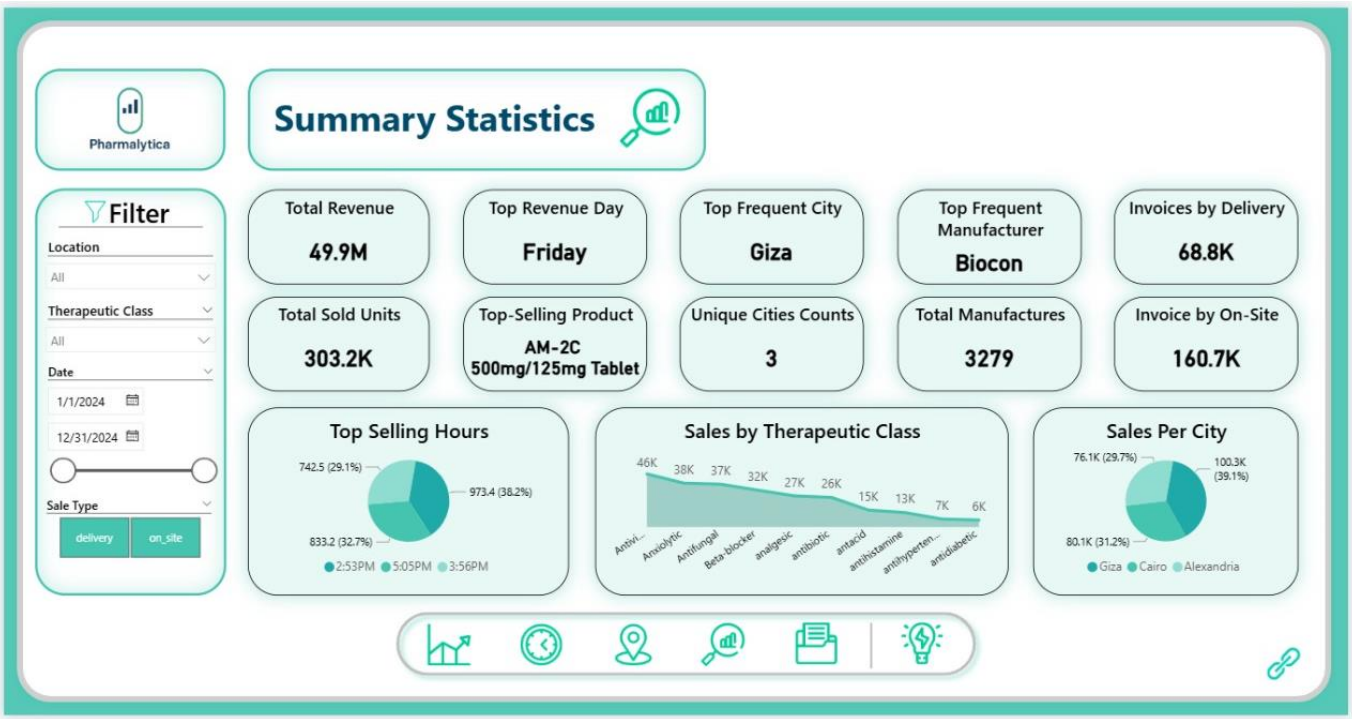

Pharmalytica
Delivery
revenue per pickup type
delivery 15.16M
on_site 34.72M
Sales per pickup type
delivery 90.47K
on_site 212.76K
AVG Sales for delivery
214.67
215.54
221.08
Alexandria
Cairo
Giza
Sales per time trend
on_site
delivery
Ja.. Fe.. M.. Ap.. May Ju.. July Au.. Se.. Oc.. No.. De..
Revenue by type and channel
product_ty.. sales_..
Drug on_site 18.26M
delivery 7.98M
Supply on_site 16.46M
delivery 7.18M
Top 5 Theraputic classes sold via delivery
10,933
9,743
9,154
8,434
7,334
Antiviral
Anxiolytic
Antifungal
Beta-bloc..
analgesic
manfacture reliance on delivery
Abbott
ABN Dru..
Cipla Ltd
Sun Pha..
Zydus C..
20.2%
20.3%
17.2%
20.6%
22.4%

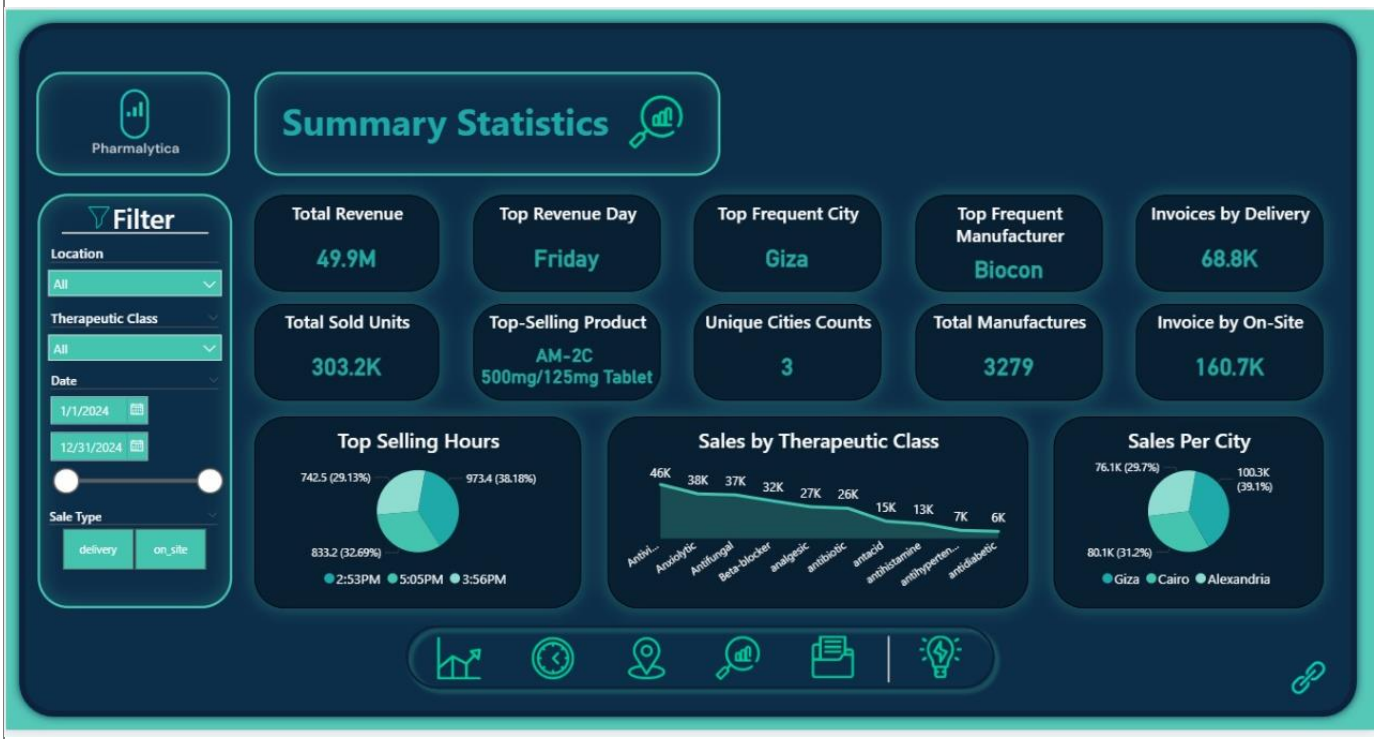
Insurance

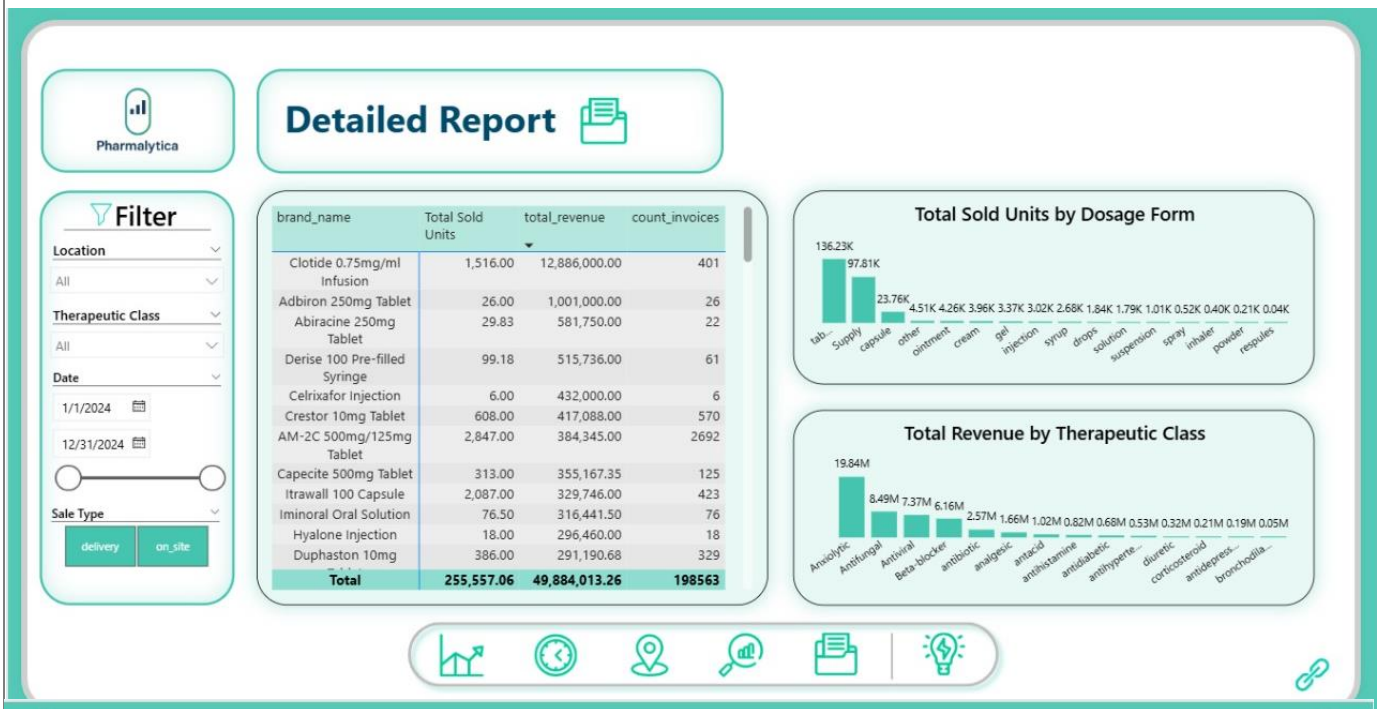




Summary Statistics


Pharmalytica
Summary Statistics
Filter
Location
All
Therapeutic Class
All
Date
1/1/2024
12/31/2024
Sale Type
delivery
on_site
Total Revenue
49.9M
Top Revenue Day
Friday
Top Frequent City
Giza
Top Frequent Manufacturer
Biocon
Invoices by Delivery
68.8K
Total Sold Units
303.2K
Top-Selling Product
AM-2C 500mg/125mg Tablet
Unique Cities Counts
3
Total Manufactures
3279
Invoice by On-Site
160.7K
Top Selling Hours
742.5 (29.1%)
973.4 (38.2%)
833.2 (32.7%)
2:53PM
5:05PM
3:56PM
Sales by Therapeutic Class
46K
38K
37K
32K
27K
26K
15K
13K
7K
6K
Antivi...
Anxiolytic
Antifungal
Beta-blocker
analgesic
antibiotic
antacid
antihistamine
antihyperten...
antidiabetic
Sales Per City
76.1K (29.7%)
100.3K (39.1%)
80.1K (31.2%)
Giza
Cairo
Alexandria


Pharmalytica
Summary Statistics
Filter
Location
All
Therapeutic Class
All
Date
1/1/2024
12/31/2024
Sale Type
delivery
on_site
Total Revenue
49.9M
Top Revenue Day
Friday
Top Frequent City
Giza
Top Frequent Manufacturer
Biocon
Invoices by Delivery
68.8K
Total Sold Units
303.2K
Top-Selling Product
AM-2C 500mg/125mg Tablet
Unique Cities Counts
3
Total Manufactures
3279
Invoice by On-Site
160.7K
Top Selling Hours
742.5 (29.13%)
973.4 (38.18%)
833.2 (32.69%)
2:53PM
5:05PM
3:56PM
Sales by Therapeutic Class
46K
38K
37K
32K
27K
26K
15K
13K
7K
6K
Antivi...
Anxiolytic
Antifungal
Beta-blocker
analgesic
antibiotic
antacid
antihistamine
antihyperten...
antidiabetic
Sales Per City
76.1K (29.7%)
100.3K (39.1%)
80.1K (31.2%)
Giza
Cairo
Alexandria


Pharmalytica
Detailed Report
Filter
Location
All
Therapeutic Class
All
Date
1/1/2024
12/31/2024
Sale Type
delivery
on_site
brand_name | Total Sold Units | total_revenue | count_invoices
Clotide 0.75mg/ml Infusion | 1,516.00 | 12,886,000.00 | 401
Adbiron 250mg Tablet | 26.00 | 1,001,000.00 | 26
Abiracine 250mg Tablet | 29.83 | 581,750.00 | 22
Derise 100 Pre-filled Syringe | 99.18 | 515,736.00 | 61
Celrixafor Injection | 6.00 | 432,000.00 | 6
Crestor 10mg Tablet | 608.00 | 417,088.00 | 570
AM-2C 500mg/125mg Tablet | 2,847.00 | 384,345.00 | 2692
Capecite 500mg Tablet | 313.00 | 355,167.35 | 125
Itrawall 100 Capsule | 2,087.00 | 329,746.00 | 423
Iminoral Oral Solution | 76.50 | 316,441.50 | 76
Hyalone Injection | 18.00 | 296,460.00 | 18
Duphaston 10mg | 386.00 | 291,190.68 | 329
Total | 255,557.06 | 49,884,013.26 | 198563
Total Sold Units by Dosage Form
136.23K 97.81K 23.76K 4.51K 4.26K 3.96K 3.37K 3.02K 2.68K 1.84K 1.79K 1.01K 0.52K 0.40K 0.21K 0.04K
tab... Supply capsule other ointment cream gel injection syrup drops solution suspension spray inhaler powder respules
Total Revenue by Therapeutic Class
19.84M 8.49M 7.37M 6.16M 2.57M 1.66M 1.02M 0.82M 0.68M 0.53M 0.32M 0.21M 0.19M 0.05M
Anxiolytic Antifungal Antiviral Beta-blocker antibiotic analgesic antacid antihistamine antidiabetic antihyperte... diuretic corticosteroid antidepress... bronchodila...


Pharmalytica
Detailed Report
Filter
Location
All
Therapeutic Class
All
Date
1/1/2024
12/31/2024
Sale Type
delivery
on_site
brand_name | Total Sold Units | total_revenue | count_invoices
Clotide 0.75mg/ml Infusion | 1,516.00 | 12,886,000.00 | 401
Adbiron 250mg Tablet | 26.00 | 1,001,000.00 | 26
Abiracine 250mg Tablet | 29.83 | 581,750.00 | 22
Derise 100 Pre-filled Syringe | 99.18 | 515,736.00 | 61
Celrixafor Injection | 6.00 | 432,000.00 | 6
Crestor 10mg Tablet | 608.00 | 417,088.00 | 570
AM-2C 500mg/125mg Tablet | 2,847.00 | 384,345.00 | 2692
Capecite 500mg Tablet | 313.00 | 355,167.35 | 125
Itrawall 100 Capsule | 2,087.00 | 329,746.00 | 423
Iminoral Oral Solution | 76.50 | 316,441.50 | 76
Hyalone Injection | 18.00 | 296,460.00 | 18
Duphaston 10mg | 386.00 | 291,190.68 | 329
Total | 255,557.06 | 49,884,013.26 | 198563
Total Sold Units by Dosage Form
136.23K 97.81K 23.76K 4.51K 4.26K 3.96K 3.37K 3.02K 2.68K 1.84K 1.79K 1.01K 0.52K 0.40K 0.21K 0.04K
tab... Supply capsule other ointment cream gel injection syrup drops solution suspension spray inhaler powder respules
Total Revenue by Therapeutic Class
19.84M 8.49M 7.37M 6.16M 2.57M 1.66M 1.02M 0.82M 0.68M 0.53M 0.32M 0.21M 0.19M 0.05M
Anxiolytic Antifungal Antiviral Beta-blocker antibiotic analgesic antacid antihistamine antidiabetic antihyperte... diuretic corticosteroid antidepress... bronchodila...