MedSynora DW – A Comprehensive Synthetic Hospital Patient Data Warehouse

Group No:15

Higher National Diploma in Software Engineering 24.2F

Datawarehouse & Business Intelligence

Project Report

Submitted By:

Index Number	Name	Contribution
COHNDSE242F-041	Ilma M H F	20%
COHNDSE242F-042	Azward M A	20%
COHNDSE242F-092	Ishini B G U	20%
COHNDSE242F-104	Jayamanne S D	20%
COHNDSE242F-114	Chathama M A L	20%

School of Computing and Engineering

National Institute of Business Management

Colombo 7

Contents

Chapter 1: Introduction

1.1 Overview of the Business Domain

MedSynora DW is a corporate synthetic hospital data warehouse for the emulation of realistic clinical and administrative activities in a huge hospital. The dataset is a fine-grained health care activity, attention is emphasized on 'patient journey'. It comprises data of patient visits, diagnoses, labs, vitals, treatments, procedures or costs and spans throughout the year 2024.

This data warehouse further includes patient-centered activities, such as admissions, doctor encounters, room utilization, insurance status, and chronic diseases. The dataset is available on Kaggle and has been designed for data warehousing and analysis applications. It is of particular interest to data engineers, healthcare analysts and system designers who are interested in building clinical dashboards and healthcare intelligence systems. **https://www.kaggle.com/datasets/mebrar21/medsynora-dw**

1.2 Objective and Purpose

Objective:

The goal of this work is to develop a database Data Warehouse to facilitate the efficient storage and retrieval of meaningful health-related data with the help of concepts, principles, and techniques of the Galaxy Schema model. The system is expected to facilitate decision-making and performance measurement in a hospital.

Purpose:

- To unify data from various sources into a single ' data warehouse '-capable of separating ' data-load ' from ' analysis ' and providing all the latest performance management facilities.
- To make data more available so that doctors, analysts and hospital administrators can actually use it.
- To be able to perform multi-dimensional data analysis using well defined fact and dimension tables.
- To monitor KPIs like cost per treatment, readmission rate, average LOS, etc.

- For reporting and strategic planning, to have similar and standardized data throughout the campus.

1.3 Key Deliverables

1. Data Warehouse
   Constructed with SQL Server Management Studio, tables created and populated from multiple sources (SQL, CSV, JSON, XML, Excel).

2. ETL Pipeline
   Created in “Apache Hop”, in charge of data ETL (extract, transform, load)

3. Dashboard
   Developed with “Power BI” for visualization of data i.e., KPI s, charts, filters, interactive document.

Chapter 2: Data Source Identification

2.1 Description of Data

The MedSynora Data set is an integrated data collection of different structured data sources in a hospital. These include:

- Patient information and encounter history.
- Doctor assignments and room details.
- Diagnoses and treatments.
- Vitals and lab tests.
- Insurance claims.
- Cost breakdowns by treatment stage.

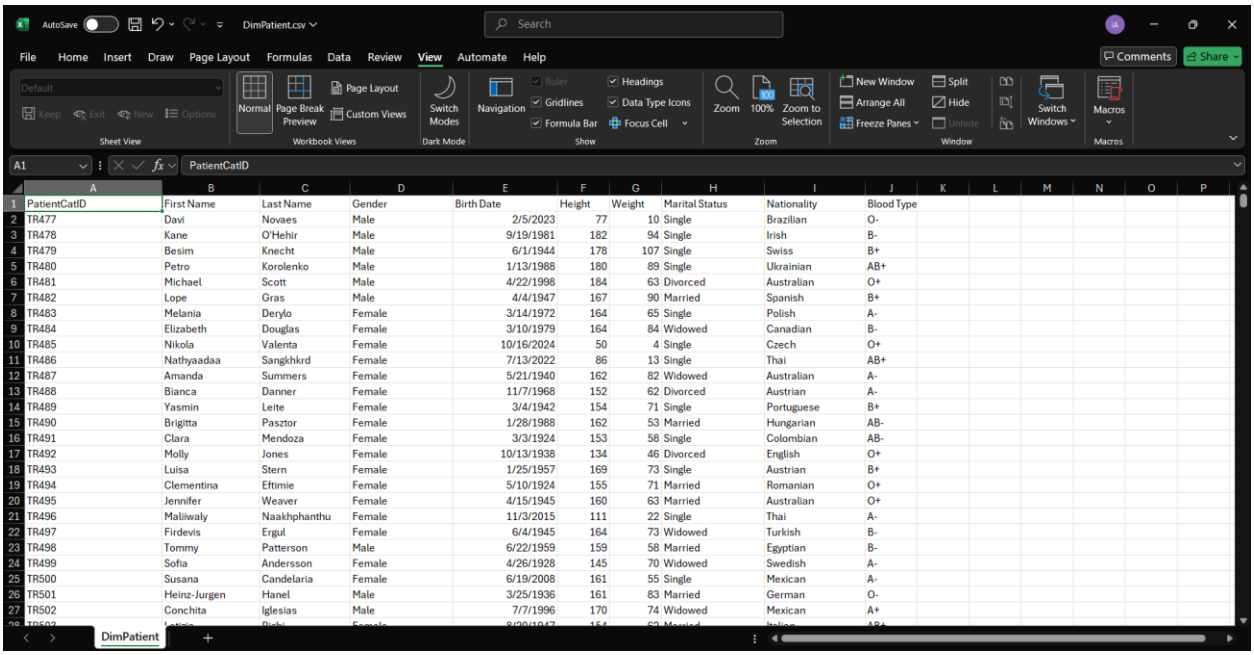
Sample attributes include:

- Encounter_ID, Patient_ID, Disease_ID, ResponsibleDoctorID, InsuranceKey, RoomKey, CheckinDate, CheckoutDate, CheckinDateKey, CheckoutDateKey, Patient_Severity_Score, RadiologyType, RadiologyProcedureCount, EndoscopyType, EndoscopyProcedureCount, CompanionPresent

2.2 Data Structures

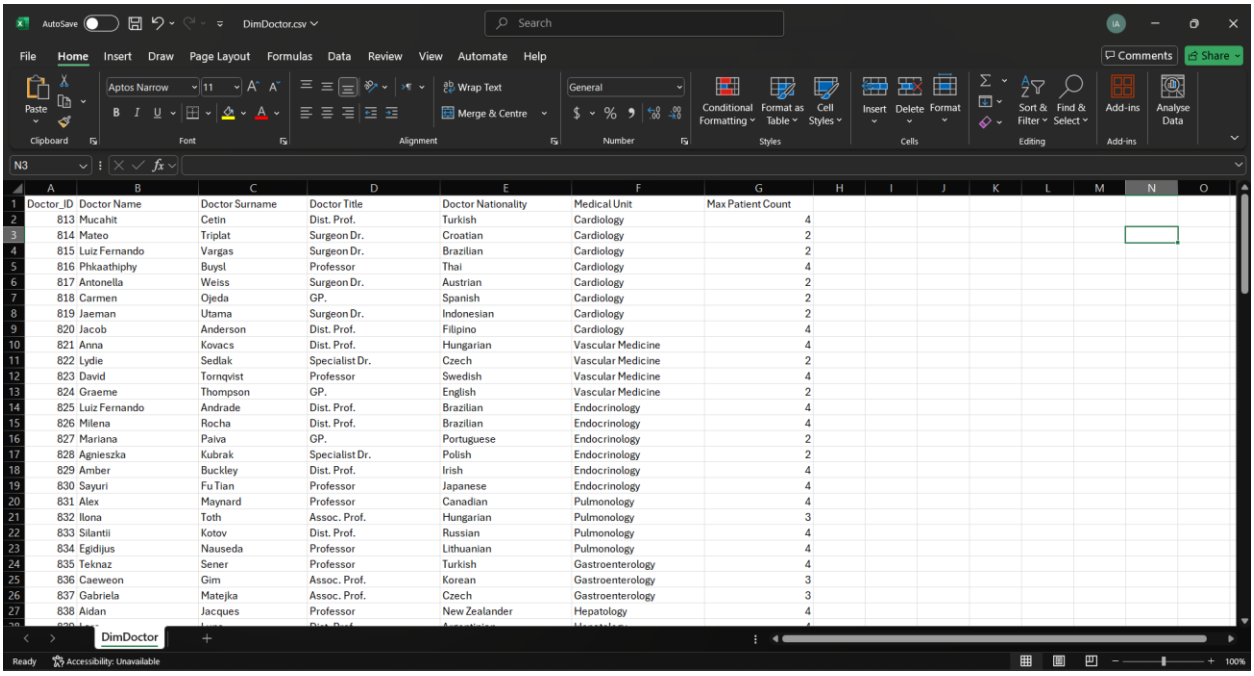
Dimension Tables (11 total)

- DimPatient (SQL)



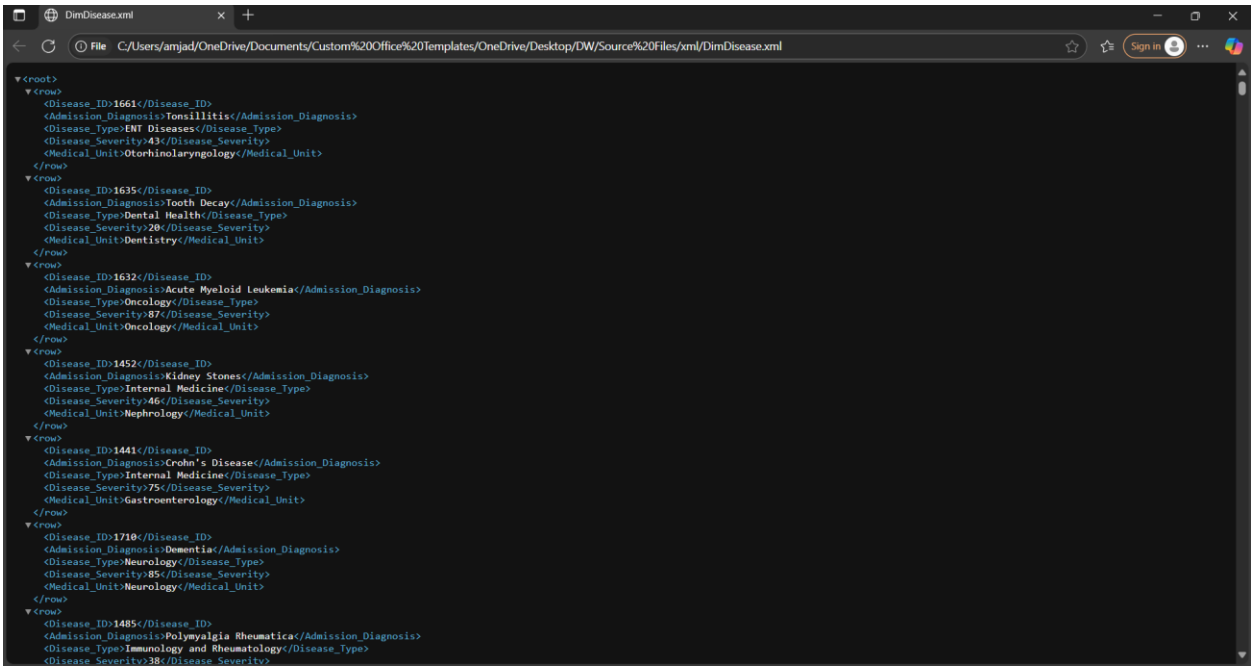
PatientCatID	First Name	Last Name	Gender	Birth Date	Height	Weight	Marital Status	Nationality	Blood Type
TR477	Davi	Novaes	Male	2/5/2023	77	10	Single	Brazilian	O-
TR478	Kane	O'Hehir	Male	9/19/1981	182	94	Single	Irish	B-
TR479	Besim	Knecht	Male	6/1/1944	178	107	Single	Swiss	B+
TR480	Petro	Korolenko	Male	1/13/1988	180	89	Single	Ukrainian	AB+
TR481	Michael	Scott	Male	4/22/1998	184	63	Divorced	Australian	O+
TR482	Lope	Gras	Male	4/4/1947	167	90	Married	Spanish	B+
TR483	Melania	Derylo	Female	3/14/1972	164	65	Single	Polish	A-
TR484	Elizabeth	Douglas	Female	3/10/1979	164	84	Widowed	Canadian	B-
TR485	Nikola	Valenta	Female	10/16/2024	50	4	Single	Czech	O+
TR486	Nathyaadaa	Sangkhkrd	Female	7/13/2022	86	13	Single	Thai	AB+
TR487	Amanda	Summers	Female	5/21/1940	162	82	Widowed	Australian	A-
TR488	Bianca	Danner	Female	11/7/1968	152	62	Divorced	Austrian	A-
TR489	Yasmin	Leite	Female	3/4/1942	154	71	Single	Portuguese	B+
TR490	Brigitta	Pasztor	Female	1/28/1988	162	53	Married	Hungarian	AB-
TR491	Clara	Mendoza	Female	3/3/1924	153	58	Single	Colombian	AB-
TR492	Molly	Jones	Female	10/13/1938	134	46	Divorced	English	O+
TR493	Luisa	Stern	Female	1/25/1957	169	73	Single	Austrian	B+
TR494	Clementina	Eftimie	Female	5/10/1924	155	71	Married	Romanian	O+
TR495	Jennifer	Weaver	Female	4/15/1945	160	63	Married	Australian	O+
TR496	Maliiwaly	Naakhphanthu	Female	11/3/2015	111	22	Single	Thai	A-
TR497	Firdevis	Ergul	Female	6/4/1945	164	73	Widowed	Turkish	B-
TR498	Tommy	Patterson	Male	6/22/1959	159	58	Married	Egyptian	B-
TR499	Sofia	Andersson	Female	4/26/1928	145	70	Widowed	Swedish	A-
TR500	Susana	Candelaria	Female	6/19/2008	161	55	Single	Mexican	A-
TR501	Heinz-Jurgen	Hanel	Male	3/25/1936	161	83	Married	German	O-
TR502	Conchita	Iglesias	Male	7/7/1996	170	74	Widowed	Mexican	A+

- DimDoctor (CSV)

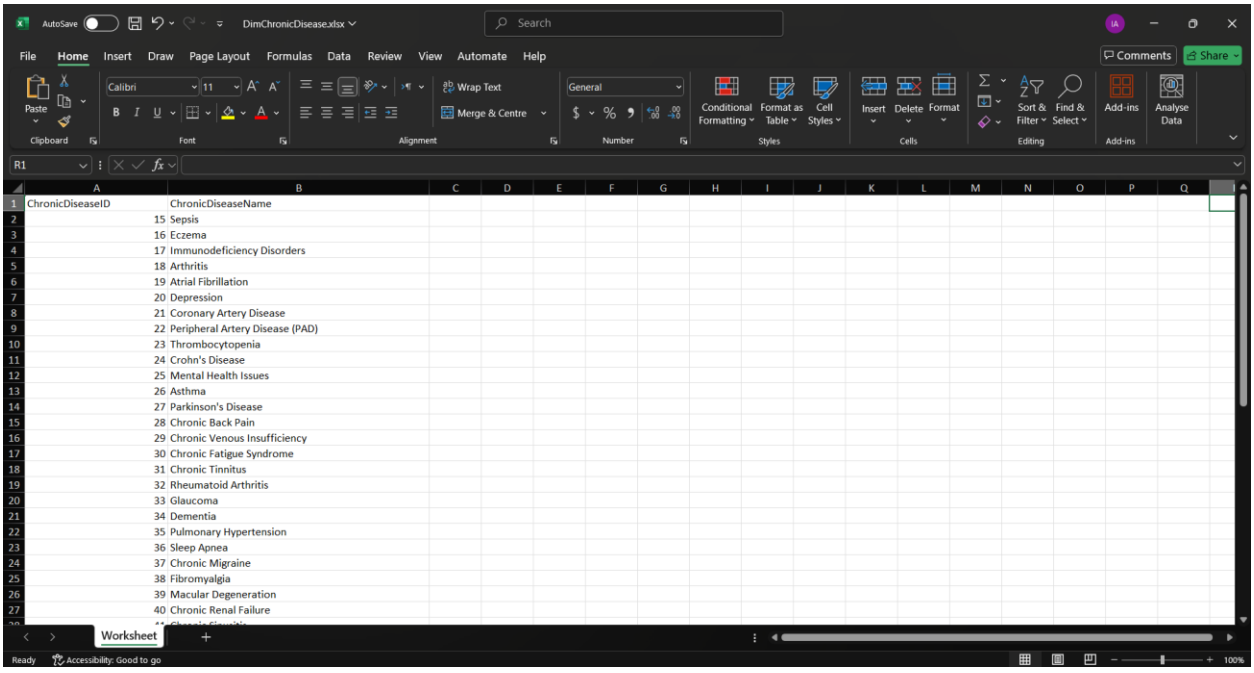


Doctor_ID	Doctor Name	Doctor Surname	Doctor Title	Doctor Nationality	Medical Unit	Max Patient Count
813	Mucahit	Cetin	Dist. Prof.	Turkish	Cardiology	4
814	Mateo	Triplat	Surgeon Dr.	Croatian	Cardiology	2
815	Luiz Fernando	Vargas	Surgeon Dr.	Brazilian	Cardiology	2
816	Phkaathiphy	Buysl	Professor	Thai	Cardiology	4
817	Antonella	Weiss	Surgeon Dr.	Austrian	Cardiology	2
818	Carmen	Ojeda	GP.	Spanish	Cardiology	2
819	Jaeman	Utama	Surgeon Dr.	Indonesian	Cardiology	2
820	Jacob	Anderson	Dist. Prof.	Filipino	Cardiology	4
821	Anna	Kovacs	Dist. Prof.	Hungarian	Vascular Medicine	4
822	Lydie	Sedlak	Specialist Dr.	Czech	Vascular Medicine	2
823	David	Tornqvist	Professor	Swedish	Vascular Medicine	4
824	Graeme	Thompson	GP.	English	Vascular Medicine	2
825	Luiz Fernando	Andrade	Dist. Prof.	Brazilian	Endocrinology	4
826	Milena	Rocha	Dist. Prof.	Brazilian	Endocrinology	4
827	Mariana	Paiva	GP.	Portuguese	Endocrinology	2
828	Agnieszka	Kubrak	Specialist Dr.	Polish	Endocrinology	2
829	Amber	Buckley	Dist. Prof.	Irish	Endocrinology	4
830	Sayuri	Fu Tian	Professor	Japanese	Endocrinology	4
831	Alex	Maynard	Professor	Canadian	Pulmonology	4
832	Ilona	Toth	Assoc. Prof.	Hungarian	Pulmonology	3
833	Silantii	Kotov	Dist. Prof.	Russian	Pulmonology	4
834	Egidijus	Nauseda	Professor	Lithuanian	Pulmonology	4
835	Teknaz	Sener	Professor	Turkish	Gastroenterology	4
836	Caeweon	Gim	Assoc. Prof.	Korean	Gastroenterology	3
837	Gabriela	Matejka	Assoc. Prof.	Czech	Gastroenterology	3
838	Aidan	Jacques	Professor	New Zealander	Hepatology	4

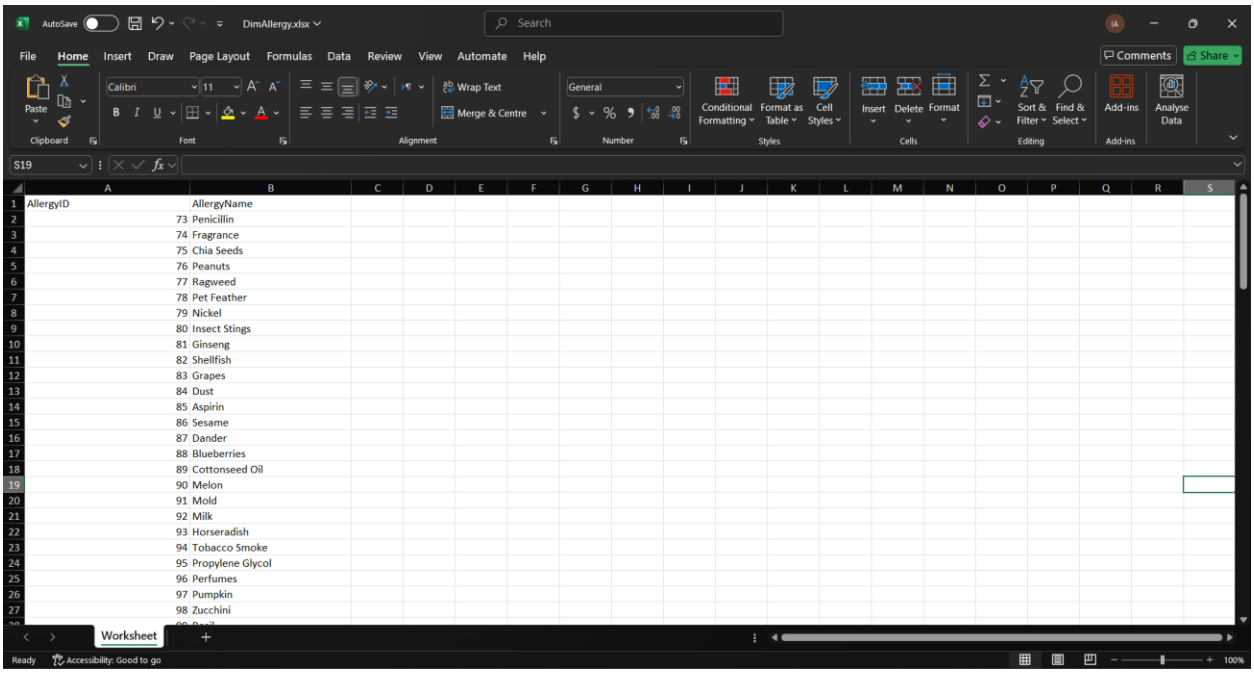
- DimDisease (XML)



- DimChronicDisease (Excel)

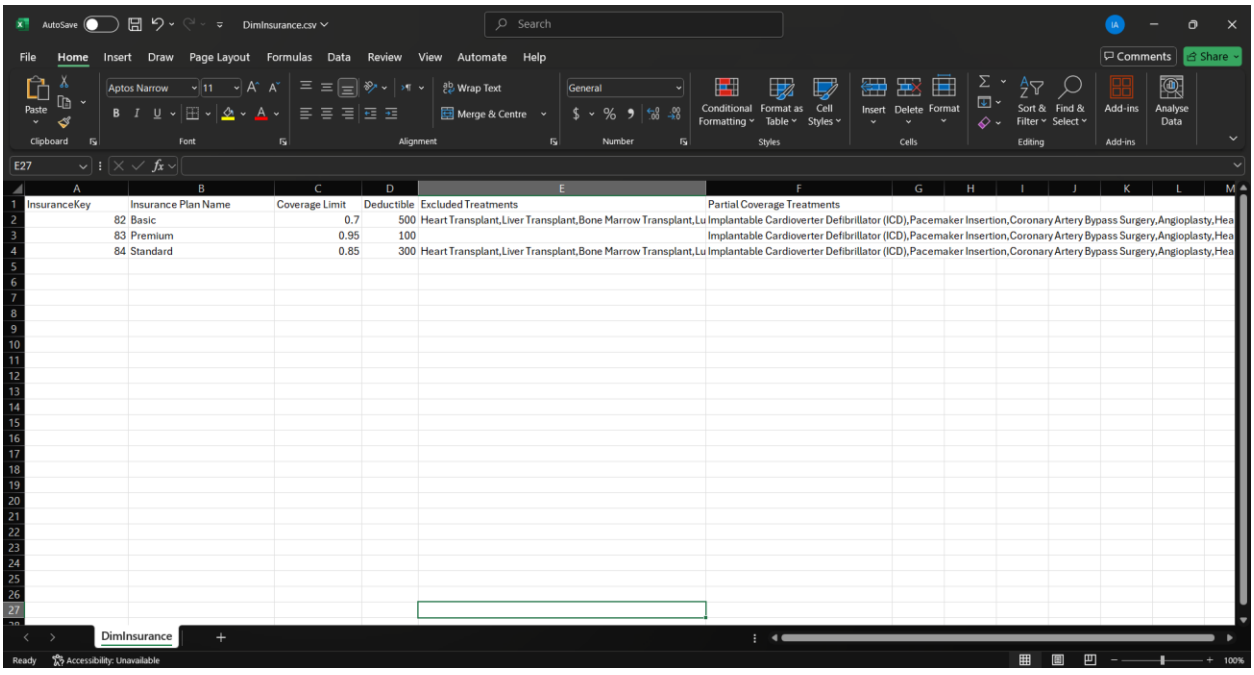


- DimAllergy (Excel)



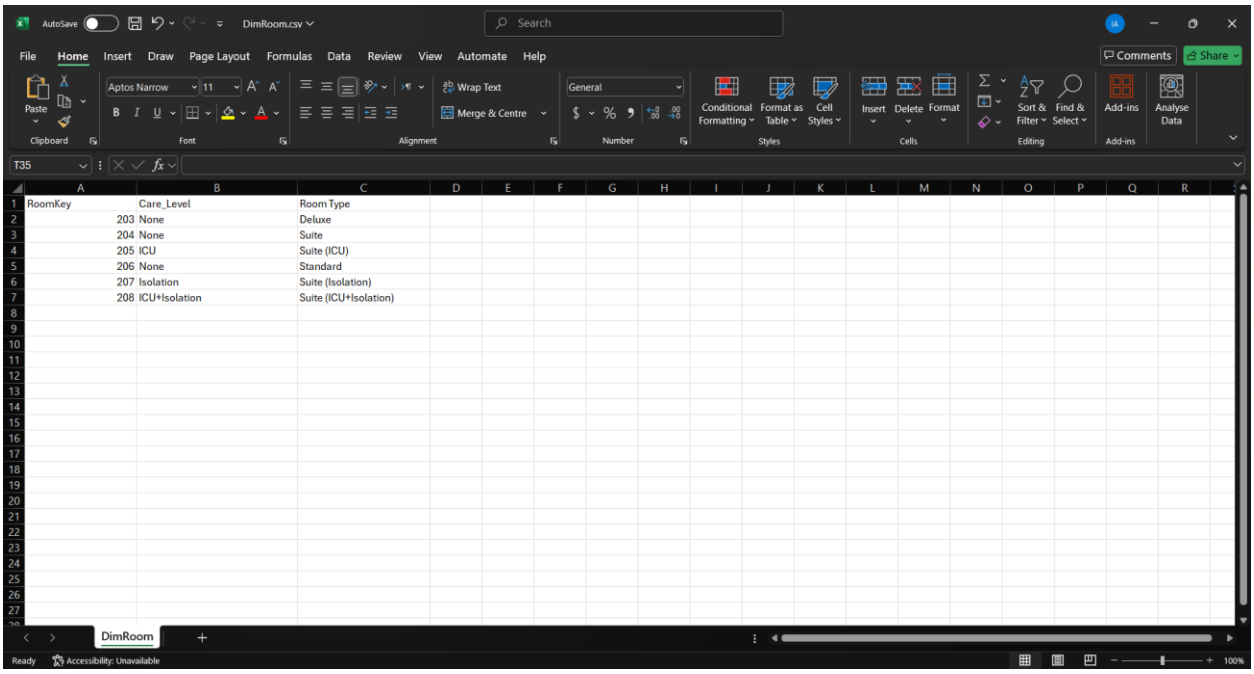
AllergyID	AllergyName
73	Penicillin
74	Fragrance
75	Chia Seeds
76	Peanuts
77	Ragweed
78	Pet Feather
79	Nickel
80	Insect Stings
81	Ginseng
82	Shellfish
83	Grapes
84	Dust
85	Aspirin
86	Sesame
87	Dander
88	Blueberries
89	Cottonseed Oil
90	Melon
91	Mold
92	Milk
93	Horseradish
94	Tobacco Smoke
95	Propylene Glycol
96	Perfumes
97	Pumpkin
98	Zucchini

- DimInsurance (CSV)



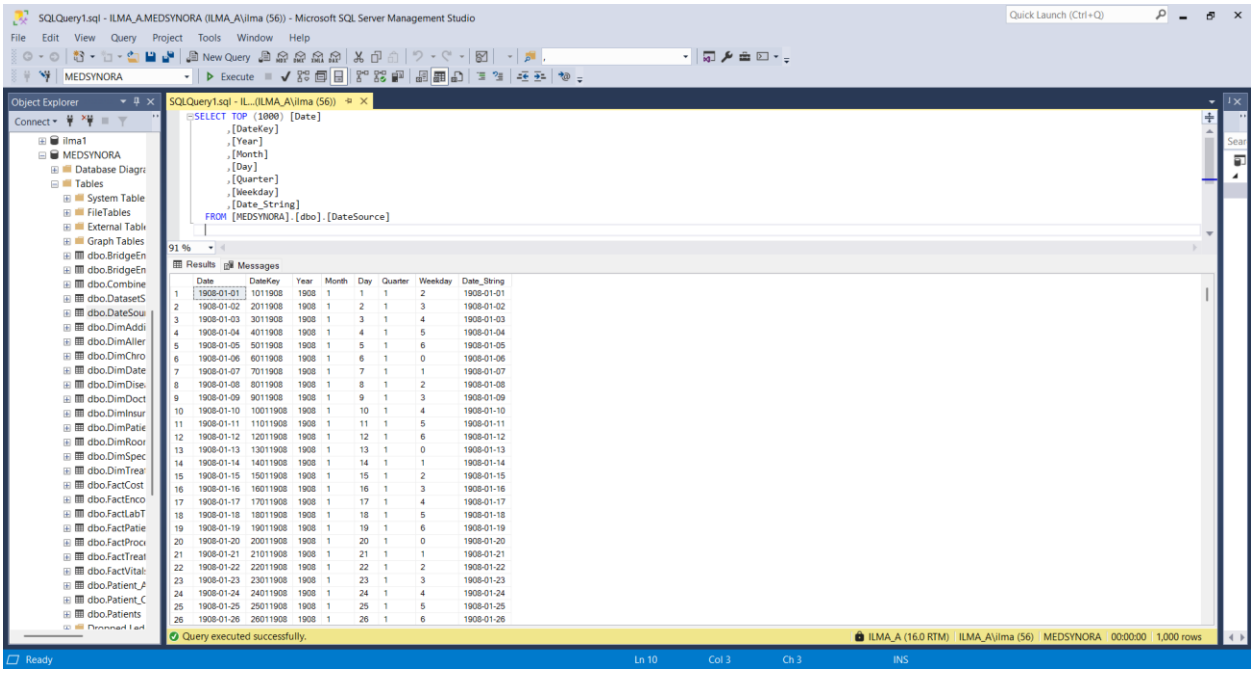
InsuranceKey	Insurance Plan Name	Coverage Limit	Deductible	Excluded Treatments	Partial Coverage Treatments
82	Basic	0.7	500	Heart Transplant,Liver Transplant,Bone Marrow Transplant,Lu	Implantable Cardioverter Defibrillator (ICD),Pacemaker Insertion,Coronary Artery Bypass Surgery,Angioplasty,Hea
83	Premium	0.95	100		Implantable Cardioverter Defibrillator (ICD),Pacemaker Insertion,Coronary Artery Bypass Surgery,Angioplasty,Hea
84	Standard	0.85	300	Heart Transplant,Liver Transplant,Bone Marrow Transplant,Lu	Implantable Cardioverter Defibrillator (ICD),Pacemaker Insertion,Coronary Artery Bypass Surgery,Angioplasty,Hea

- DimRoom (CSV)



RoomKey	Care_Level	Room Type
203	None	Deluxe
204	None	Suite
205	ICU	Suite (ICU)
206	None	Standard
207	Isolation	Suite (Isolation)
208	ICU+Isolation	Suite (ICU+Isolation)

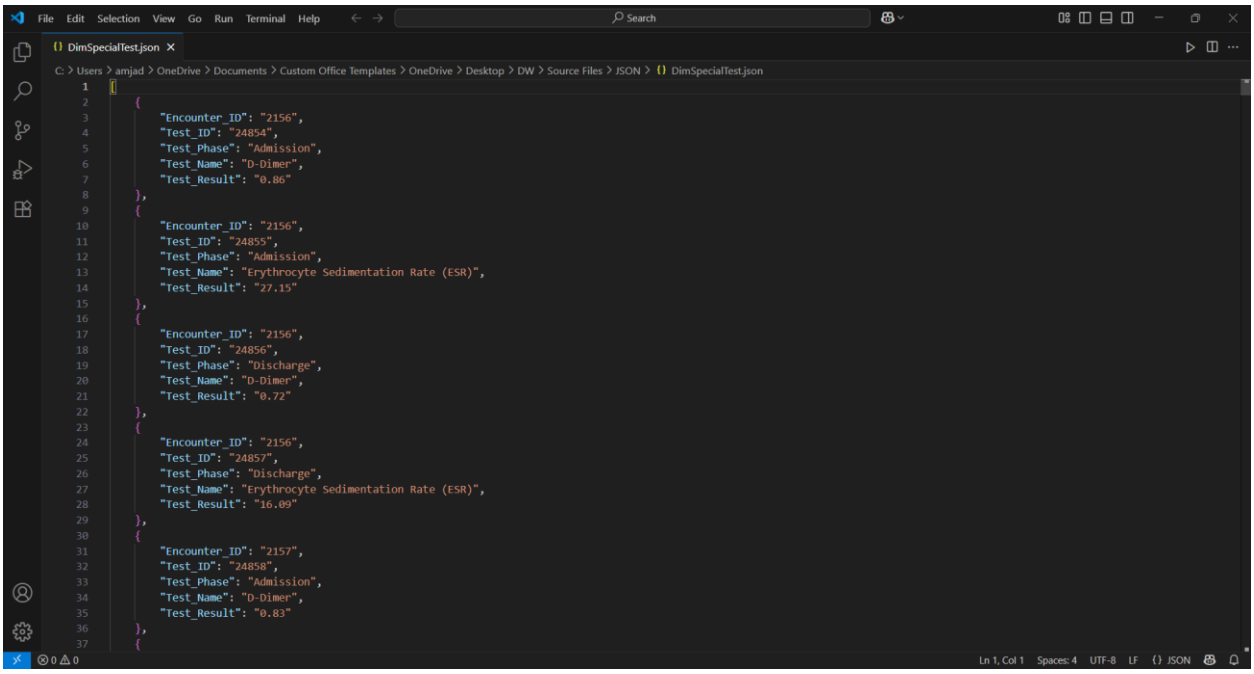
- DimDate (SQL)



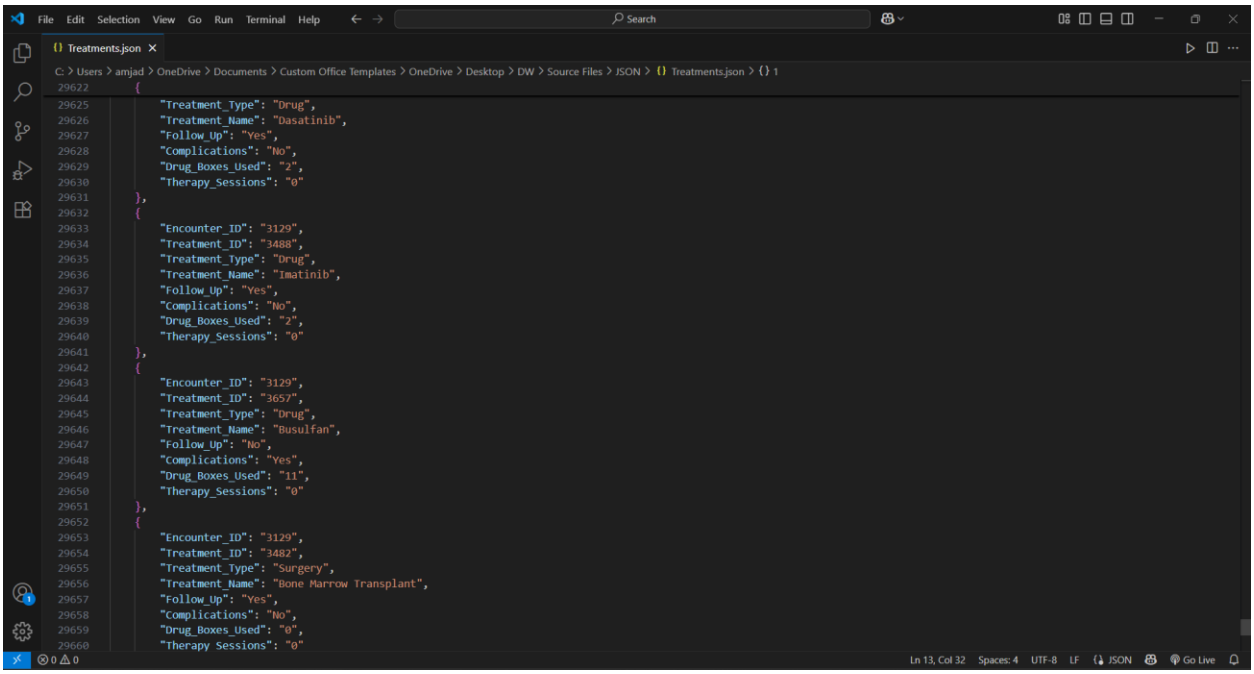
```
SELECT TOP (1000) [Date]
      ,[DateKey]
      ,[Year]
      ,[Month]
      ,[Day]
      ,[Quarter]
      ,[Weekday]
      ,[Date_String]
  FROM [MEDSYNORA].[dbo].[DateSource]
```

Date	DateKey	Year	Month	Day	Quarter	Weekday	Date_String
1908-01-01	1011908	1908	1	1	1	2	1908-01-01
1908-01-02	2011908	1908	1	2	1	3	1908-01-02
1908-01-03	3011908	1908	1	3	1	4	1908-01-03
1908-01-04	4011908	1908	1	4	1	5	1908-01-04
1908-01-05	5011908	1908	1	5	1	6	1908-01-05
1908-01-06	6011908	1908	1	6	1	0	1908-01-06
1908-01-07	7011908	1908	1	7	1	1	1908-01-07
1908-01-08	8011908	1908	1	8	1	2	1908-01-08
1908-01-09	9011908	1908	1	9	1	3	1908-01-09
1908-01-10	10011908	1908	1	10	1	4	1908-01-10
1908-01-11	11011908	1908	1	11	1	5	1908-01-11
1908-01-12	12011908	1908	1	12	1	6	1908-01-12
1908-01-13	13011908	1908	1	13	1	0	1908-01-13
1908-01-14	14011908	1908	1	14	1	1	1908-01-14
1908-01-15	15011908	1908	1	15	1	2	1908-01-15
1908-01-16	16011908	1908	1	16	1	3	1908-01-16
1908-01-17	17011908	1908	1	17	1	4	1908-01-17
1908-01-18	18011908	1908	1	18	1	5	1908-01-18
1908-01-19	19011908	1908	1	19	1	6	1908-01-19
1908-01-20	20011908	1908	1	20	1	0	1908-01-20
1908-01-21	21011908	1908	1	21	1	1	1908-01-21
1908-01-22	22011908	1908	1	22	1	2	1908-01-22
1908-01-23	23011908	1908	1	23	1	3	1908-01-23
1908-01-24	24011908	1908	1	24	1	4	1908-01-24
1908-01-25	25011908	1908	1	25	1	5	1908-01-25
1908-01-26	26011908	1908	1	26	1	6	1908-01-26

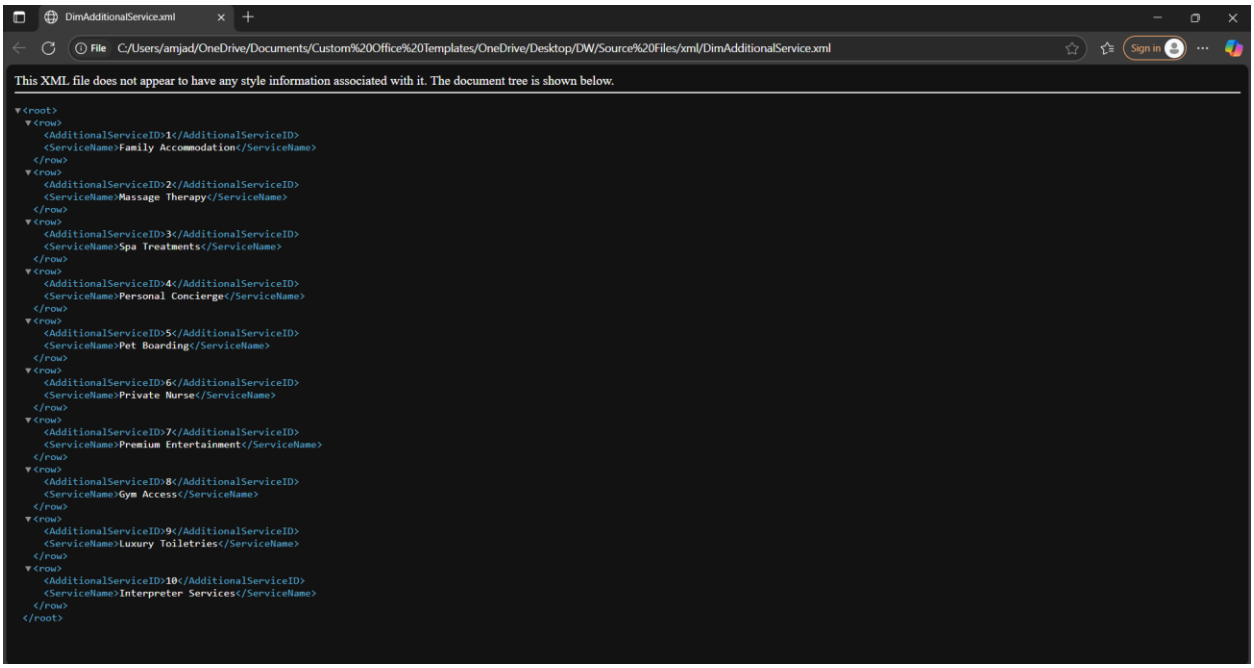
- DimSpecialTest (JSON)



- DimTreatment (JSON)

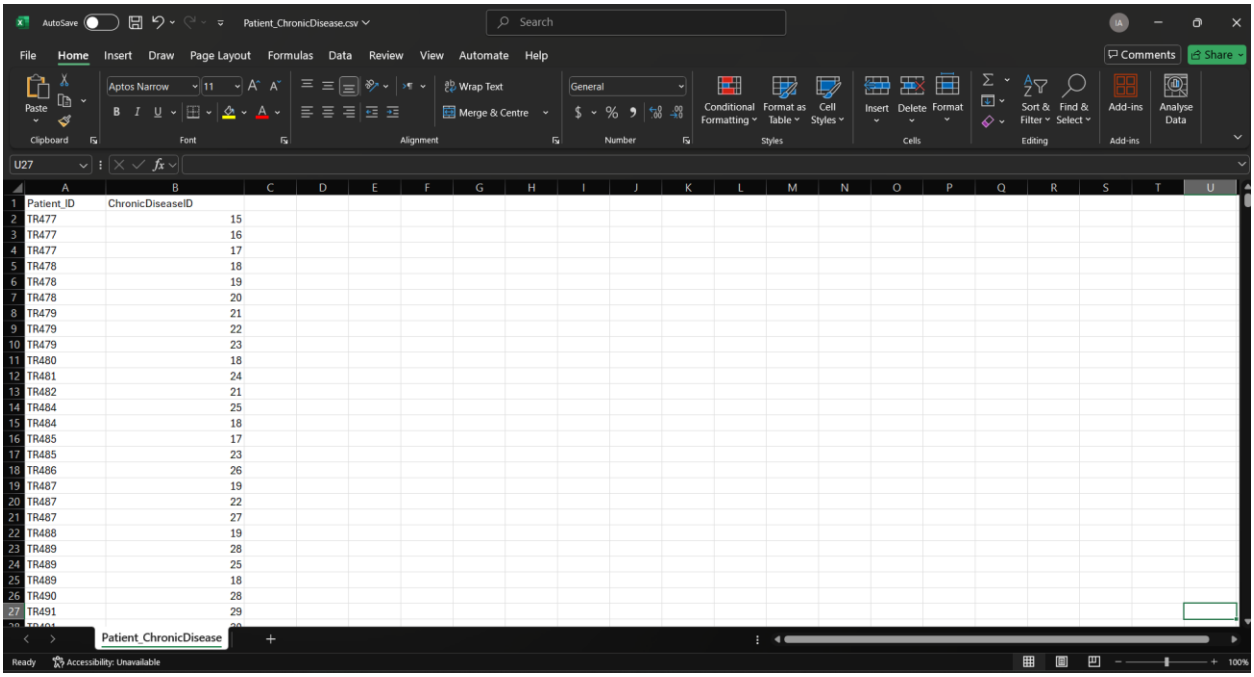


- DimAdditionalService (XML)



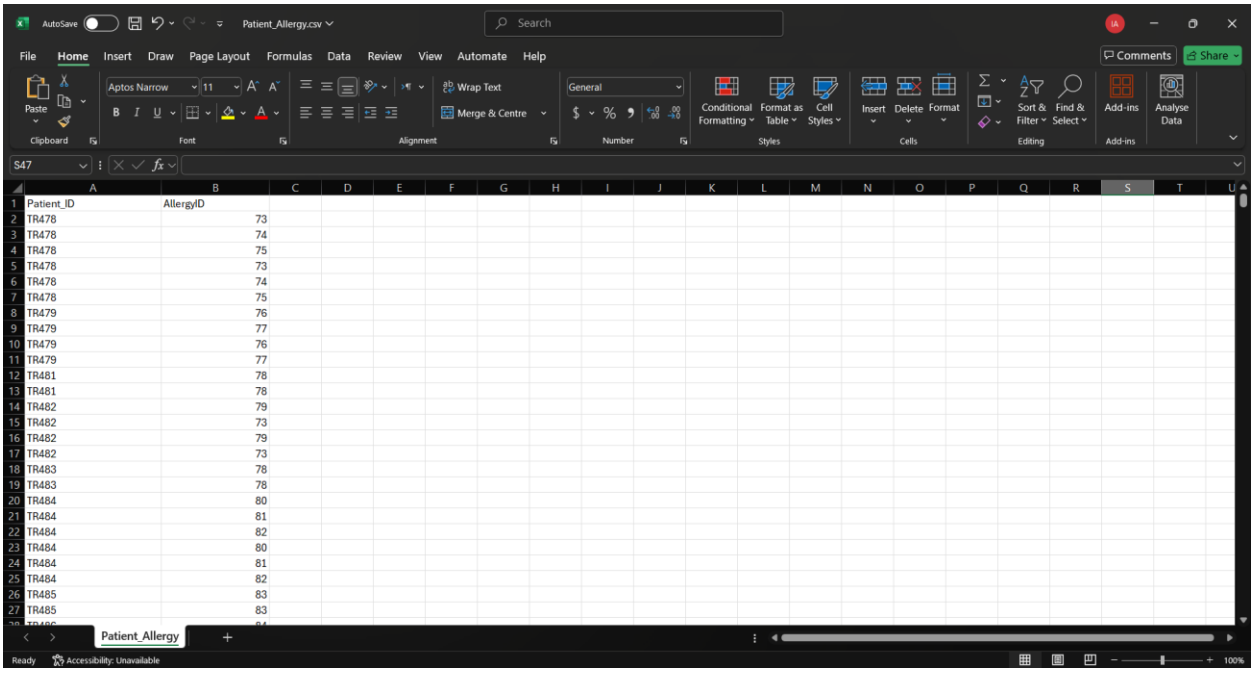
Bridge Tables (4 total, CSV format):

- Patient_ChronicDisease



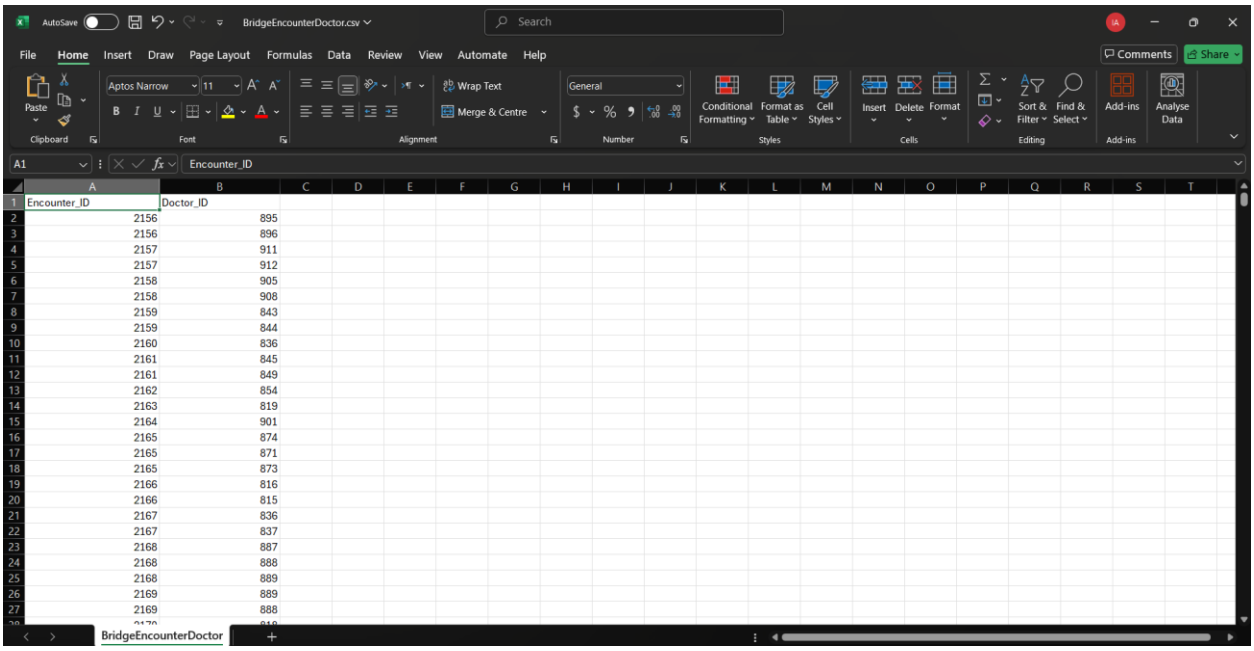
Patient_ID	ChronicDiseaseID
TR477	15
TR477	16
TR477	17
TR478	18
TR478	19
TR478	20
TR479	21
TR479	22
TR479	23
TR480	18
TR481	24
TR482	21
TR484	25
TR484	18
TR485	17
TR485	23
TR486	26
TR487	19
TR487	22
TR487	27
TR488	19
TR489	28
TR489	25
TR489	18
TR490	28
TR491	29

- Patient_Allergy



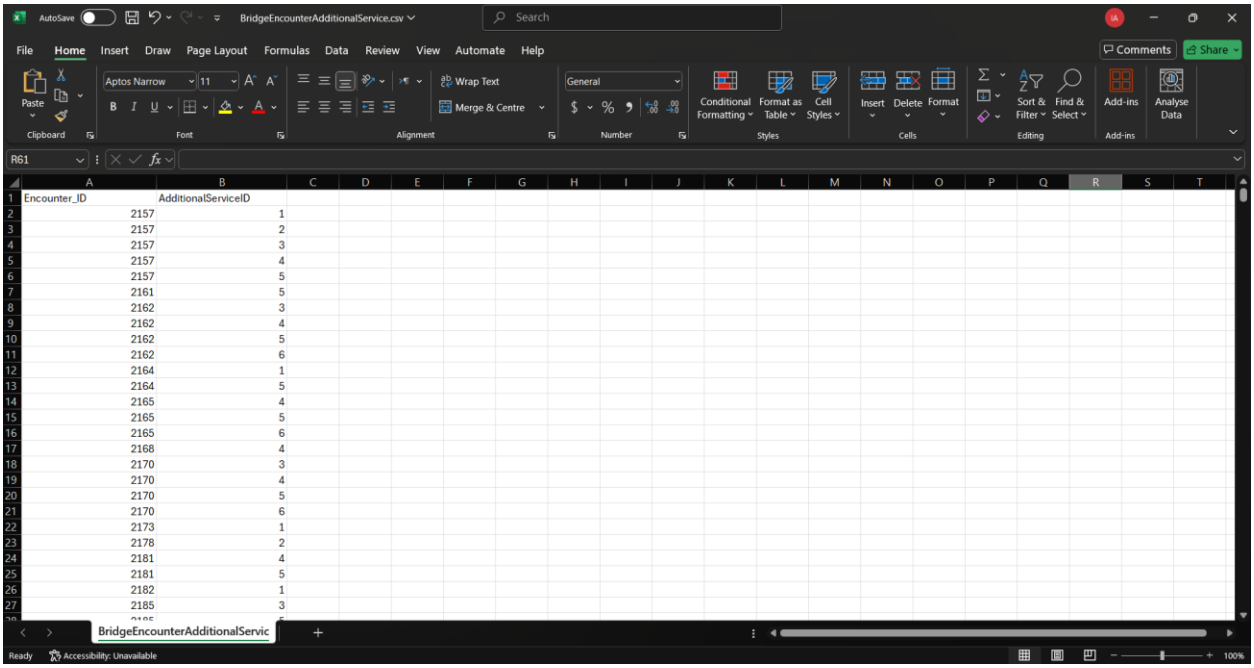
Patient_ID	AllergyID
TR478	73
TR478	74
TR478	75
TR478	73
TR478	74
TR478	75
TR479	76
TR479	77
TR479	76
TR479	77
TR481	78
TR481	78
TR482	79
TR482	73
TR482	79
TR482	73
TR483	78
TR483	78
TR484	80
TR484	81
TR484	82
TR484	80
TR484	81
TR484	82
TR485	83
TR485	83

- BridgeEncounterDoctor



Encounter_ID	Doctor_ID
2156	895
2156	896
2157	911
2157	912
2158	905
2158	908
2159	843
2159	844
2160	836
2161	845
2161	849
2162	854
2163	819
2164	901
2165	874
2165	871
2165	873
2166	816
2166	815
2167	836
2167	837
2168	887
2168	888
2168	889
2169	889
2169	888

- BridgeEncounterAdditionalService



Encounter_ID	AdditionalServiceID
2157	1
2157	2
2157	3
2157	4
2157	5
2161	5
2162	3
2162	4
2162	5
2162	6
2164	1
2164	5
2165	4
2165	5
2165	6
2168	4
2170	3
2170	4
2170	5
2170	6
2173	1
2178	2
2181	4
2181	5
2182	1
2185	3

Fact Tables (7 total):

- FactEncounter
- FactTreatment
- FactLabTests
- FactVitals
- FactCost
- FactProcedures
- FactPatientHealthService

2.3 Justification for Data Source Selection

The diversity of data formats (SQL, CSV, JSON, XML, Excel) mirrors the complexity of healthcare systems in real life, where data comes from EHRs, insurance systems, lab results and financial systems. This combination makes the warehouse powerful, practical, and scalable.

Chapter 3: Data Warehouse Design

3.1 Galaxy Schema Overview

The warehouse design follows a **Galaxy Schema** (also called a fact constellation schema), which supports multiple fact tables sharing dimension tables. This model is ideal for healthcare data, where different types of measures (e.g., encounters, treatments, costs) share common dimensions like patient, date, doctor, and disease.

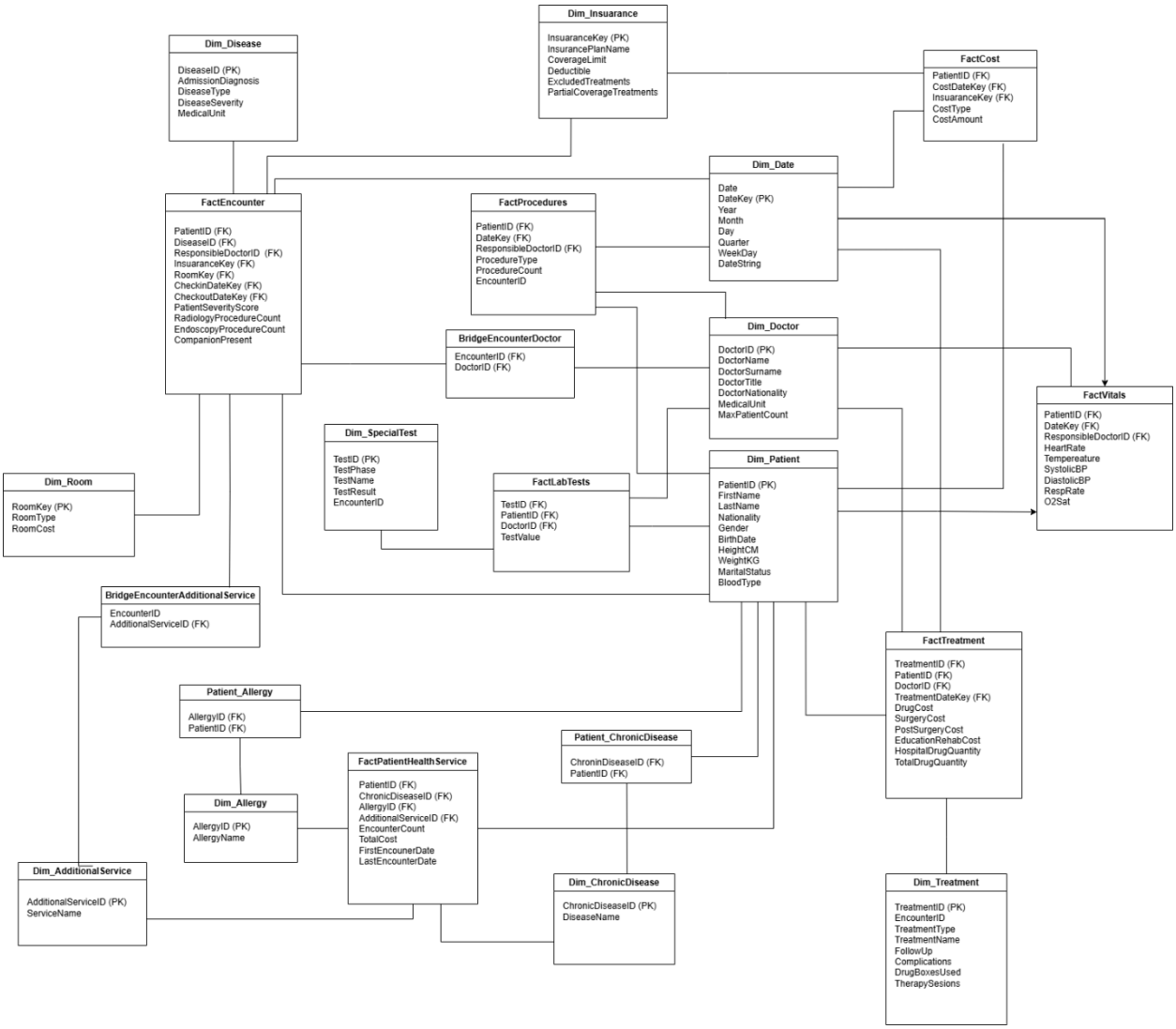
3.2 Fact Tables: Grain and Business Process

Table Name	Grain	Business Process
FactEncounter	One record per patient admission	Monitor check-in/out, when patients were in a room, insurance, doctor, and disease
FactTreatment	One record per treatment event	Records drug use, surgery price, therapy visits
FactLabTests	One record is generated for each laboratory test performed for each patient.	Tracks test name, result, and phase
FactVitals	One record for each vital sign per date	Records heart rate, temperature, O2 levels, BP
FactCost	One record for each cost entry and patient and day	Tracks billing costs across categories
FactProcedures	One record per procedure event	Monitors radiology, endoscopy and other procedures
FactPatientHealthService	One record per patient	Summarizes chronic diseases, allergies, services

The fact tables enable stakeholders (decision makers) to evaluate hospital performance from different dimensions: clinical, operational, and financial.

3.3 Data Warehouse Design

In terms of data warehouse design, the following "Galaxy Schema" can be shown.



After the Galaxy Schema Design, the following database schema has designed using "Microsoft SQL Server Management Studio (SSMS)." The following queries are used to create the mentioned database schema.

1. Database Creation

```
CREATE TABLE MEDSYNORA
USE MEDSYNORA
```

2. Dimension Table Creation

```
-- Table 1: DimDate
CREATE TABLE DimDate (
    Date DATE,
    DateKey INT PRIMARY KEY,
    Year INT,
    Month INT,
    Day INT,
    Quarter INT,
    Weekday NVARCHAR(10),
    Date_String NVARCHAR(50)
);


-- Table 2: DimPatient
CREATE TABLE DimPatient (
    Patient_ID NVARCHAR(100) PRIMARY KEY,
    FirstName NVARCHAR(100),
    LastName NVARCHAR(100),
    Nationality NVARCHAR(100),
    Gender NVARCHAR(10),
    BirthDate DATE,
    HeightCM INT,
    WeightKG INT,
    MaritalStatus NVARCHAR(20),
    BloodType NVARCHAR(5)
);

-- Table 3: DimDoctor
CREATE TABLE DimDoctor (
    Doctor_ID INT PRIMARY KEY,
    DoctorName NVARCHAR(100),
    DoctorSurname NVARCHAR(100),
    DoctorTitle NVARCHAR(50),
    DoctorNationality NVARCHAR(50),
    MedicalUnit NVARCHAR(100),
    MaxPatientCount NVARCHAR(100)
);
```

```sql
-- Table 4: DimDisease
CREATE TABLE DimDisease (
    Disease_ID INT PRIMARY KEY,
    AdmissionDiagnosis NVARCHAR(200),
    DiseaseType NVARCHAR(100),
    DiseaseSeverity NVARCHAR(50),
    MedicalUnit NVARCHAR(100)
);

-- Table 5: DimInsurance
CREATE TABLE DimInsurance (
    InsuranceKey INT PRIMARY KEY,
    Insurance_Plan_Name NVARCHAR(100),
    Coverage_Limit DECIMAL(18, 2),
    Deductible DECIMAL(18, 2),
    Excluded_Treatments NVARCHAR(MAX),
    Partial_Coverage_Treatments NVARCHAR(MAX)
);

-- Table 6: DimRoom
CREATE TABLE DimRoom (
    RoomKey INT PRIMARY KEY,
    RoomType NVARCHAR(50),
    RoomCost DECIMAL(10, 2)
);

-- Table 7: DimChronicDisease
CREATE TABLE DimChronicDisease (
    ChronicDiseaseID INT PRIMARY KEY,
    DiseaseName NVARCHAR(100)
);

-- Table 8: DimAllergy
CREATE TABLE DimAllergy (
    AllergyID INT PRIMARY KEY,
    AllergyName NVARCHAR(100)
);
```

```sql
-- Table 9: DimSpecialTest
CREATE TABLE DimSpecialTest (
    Encounter_ID INT,
    Test_ID INT PRIMARY KEY,
    Test_Phase NVARCHAR(50),
    Test_Name NVARCHAR(100),
    Test_Result NVARCHAR(50)
);
use MEDSYNORA

-- Table 10: DimTreatment
CREATE TABLE DimTreatment (
    Treatment_ID INT PRIMARY KEY,
    Encounter_ID INT,
    Treatment_Type NVARCHAR(50),
    Treatment_Name NVARCHAR(100),
    Follow_Up NVARCHAR(10),
    Complications NVARCHAR(10),
    Drug_Boxes_Used INT,
    Therapy_Sessions INT
);

-- Table 11: DimAdditionalService
CREATE TABLE DimAdditionalService (
    AdditionalServiceID INT PRIMARY KEY,
    ServiceName NVARCHAR(100)
);
```

3. Bridge Table Creation

```
CREATE TABLE BridgeEncounterDoctor (
    Encounter_ID INT,
    Doctor_ID INT,
    PRIMARY KEY (Encounter_ID, Doctor_ID),
    FOREIGN KEY (Doctor_ID) REFERENCES DimDoctor(Doctor_ID)
);


CREATE TABLE Patient_ChronicDisease (
    Patient_ID  NVARCHAR(100),
    ChronicDiseaseID INT,
    PRIMARY KEY (Patient_ID, ChronicDiseaseID),
    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (ChronicDiseaseID) REFERENCES DimChronicDisease(ChronicDiseaseID)
);


CREATE TABLE Patient_Allergy (
    Patient_ID NVARCHAR(100) ,
    AllergyID INT,
    PRIMARY KEY (Patient_ID, AllergyID),
    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (AllergyID) REFERENCES DimAllergy(AllergyID)
);
CREATE TABLE BridgeEncounterAdditionalService (
    Encounter_ID INT,
    AdditionalServiceID INT,
    PRIMARY KEY (Encounter_ID, AdditionalServiceID),
    FOREIGN KEY (AdditionalServiceID) REFERENCES DimAdditionalService(AdditionalServiceID)
);
```

4. Fact Table Creation

```
-- FACT: Cost
CREATE TABLE FactCost (
    Patient_ID NVARCHAR(100),
    CostDateKey INT,
    InsuranceKey INT,
    CostType NVARCHAR(100),
    CostAmount DECIMAL(12,2),

    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (InsuranceKey) REFERENCES DimInsurance(InsuranceKey),
    FOREIGN KEY (CostDateKey) REFERENCES DimDate(DateKey)
);
```

```sql
-- FACT: Lab Tests
CREATE TABLE FactLabTests (
    Test_ID INT,
    Patient_ID NVARCHAR(100),
    Doctor_ID INT,
    Test_Value NVARCHAR(100),

    FOREIGN KEY (Test_ID) REFERENCES DimSpecialTest(Test_ID),
    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (Doctor_ID) REFERENCES DimDoctor(Doctor_ID)
);

-- FACT: Vitals
CREATE TABLE FactVitals (
    Patient_ID NVARCHAR(100),
    DateKey INT,
    HeartRate FLOAT,
    Temperature FLOAT,
    SystolicBP FLOAT,
    DiastolicBP FLOAT,
    RespRate FLOAT,
    O2Sat FLOAT,
    ResponsibleDoctorID INT,

    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (DateKey) REFERENCES DimDate(DateKey),
    FOREIGN KEY (ResponsibleDoctorID) REFERENCES DimDoctor(Doctor_ID)
);

-- FACT: Procedures
CREATE TABLE FactProcedures (
    Encounter_ID INT,
    Patient_ID NVARCHAR(100),
    ProcedureType NVARCHAR(50),
    ProcedureCount INT,
    DateKey INT,
    ResponsibleDoctorID INT,

    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (DateKey) REFERENCES DimDate(DateKey),
    FOREIGN KEY (ResponsibleDoctorID) REFERENCES DimDoctor(Doctor_ID)
);

-- FACT: Patient Health Service
CREATE TABLE FactPatientHealthService (
    Patient_ID NVARCHAR(100),
    ChronicDiseaseID INT,
    AllergyID INT,
    AdditionalServiceID INT,
    Encounter_Count INT,
    Total_Cost DECIMAL(18,2),
    First_Encounter_Date INT,
    Last_Encounter_Date INT,

    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (ChronicDiseaseID) REFERENCES DimChronicDisease(ChronicDiseaseID),
    FOREIGN KEY (AllergyID) REFERENCES DimAllergy(AllergyID),
    FOREIGN KEY (AdditionalServiceID) REFERENCES DimAdditionalService(AdditionalServiceID)
);
```

```sql
-- FACT: Encounter
CREATE TABLE FactEncounter (
    Encounter_ID INT,
    Patient_ID NVARCHAR(100),
    Disease_ID INT,
    ResponsibleDoctorID INT,
    InsuranceKey INT,
    RoomKey INT,
    CheckinDateKey INT,
    CheckoutDateKey INT,
    Patient_Severity_Score INT,
    RadiologyProcedureCount INT,
    EndoscopyProcedureCount INT,
    CompanionPresent BIT,

    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (Disease_ID) REFERENCES DimDisease(Disease_ID),
    FOREIGN KEY (ResponsibleDoctorID) REFERENCES DimDoctor(Doctor_ID),
    FOREIGN KEY (InsuranceKey) REFERENCES DimInsurance(InsuranceKey),
    FOREIGN KEY (RoomKey) REFERENCES DimRoom(RoomKey),
    FOREIGN KEY (CheckinDateKey) REFERENCES DimDate(DateKey),
    FOREIGN KEY (CheckoutDateKey) REFERENCES DimDate(DateKey)
);

-- FACT: Treatment
CREATE TABLE FactTreatment (
    Treatment_ID INT,
    Patient_ID NVARCHAR(100),
    Doctor_ID INT,
    TreatmentDateKey INT,

    Drug_Cost DECIMAL(10,2),
    Surgery_Cost DECIMAL(10,2),
    Post_Surgery_Care_Cost DECIMAL(10,2),
    Education_Rehab_Cost DECIMAL(10,2),
    Hospital_Drug_Quantity INT,
    Discharge_Drug_Quantity INT,
    Total_Drug_Quantity INT,

    FOREIGN KEY (Treatment_ID) REFERENCES DimTreatment(Treatment_ID),
    FOREIGN KEY (Patient_ID) REFERENCES DimPatient(Patient_ID),
    FOREIGN KEY (Doctor_ID) REFERENCES DimDoctor(Doctor_ID),
    FOREIGN KEY (TreatmentDateKey) REFERENCES DimDate(DateKey)
);
```

Chapter 4: ETL Process Using Apache Hop

4.1 Tools Used

Apache Hop Lightweight open-source platform for orchestrating data pipelines to build ETL-runnable workloads on a scale. It is a GUI driven tool which also supports the designing of different pipelines and different data sources.

4.2 ETL Workflow

- **Extract:** Read data from the source (SQL, CSV, XML etc) and store the raw data in one place (Persistent storage).

  Step 1: Extract dimension data from sources (SQL, CSV, XML, etc.)

  Step 2-4: Load Bridge tables (Patient_ChronicDisease, Patient_Allergy, Patient_AdditionalService…...)

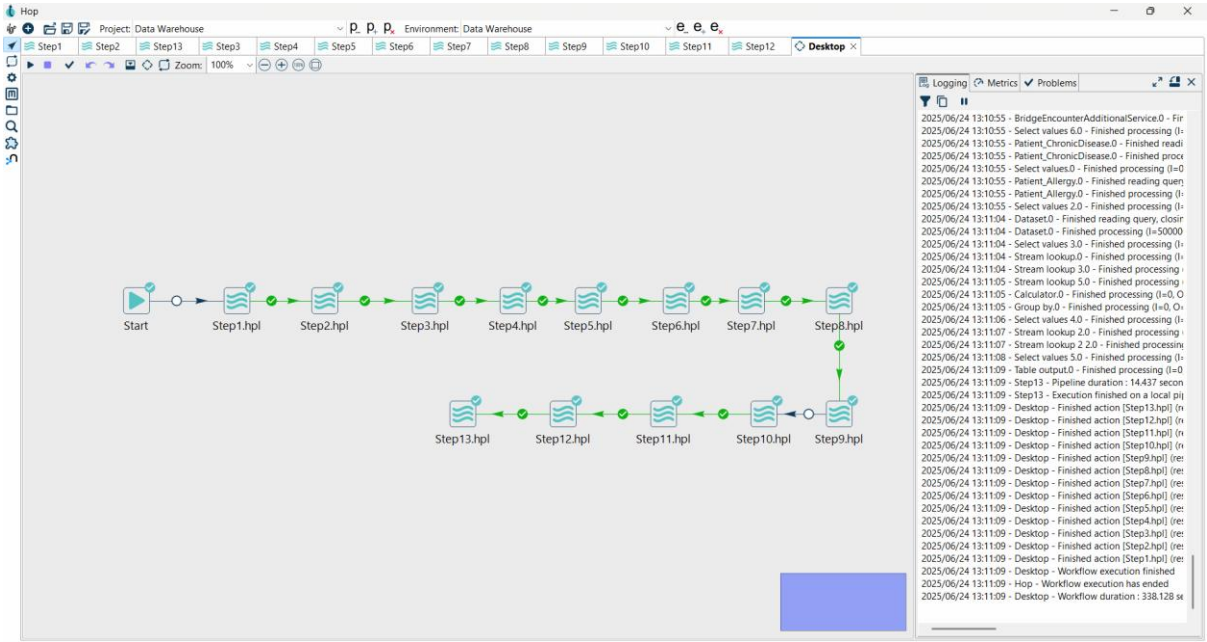
  Step 5-6: Create bill and service assignment bridge tables by the doctor and service on encounter.

  Step 7-13: Provide data into fact tables in sequence, along with keys and data range/integrity.

- **Transform:** Techniques such as data cleaning, aggregation, validation using ETL transforms were done to ensure the quality of data.

- **Load:** Transformed data were loaded to SQL Server Management Studio (SSMS) which is selected as a Database Schema.

Workflow -



PIPLINES

STEP 1 –

DimDate

DimPatient

DimDoctor

DimDisease

DimInsurance

DimRoom

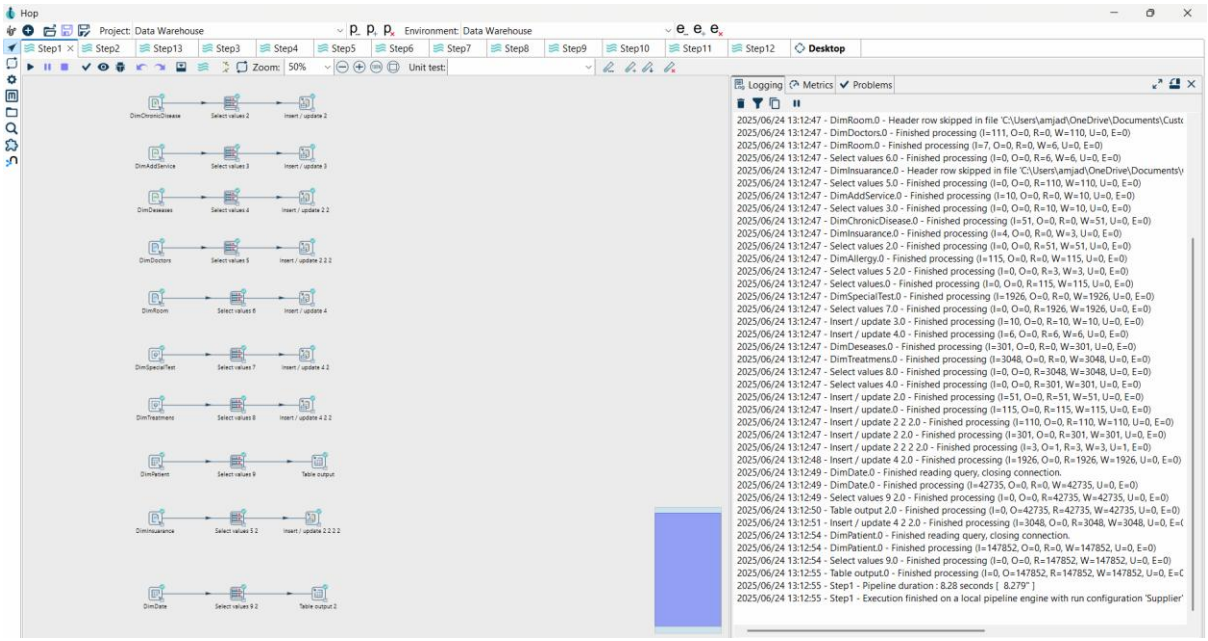
DimChronicDisease

DimAllergy

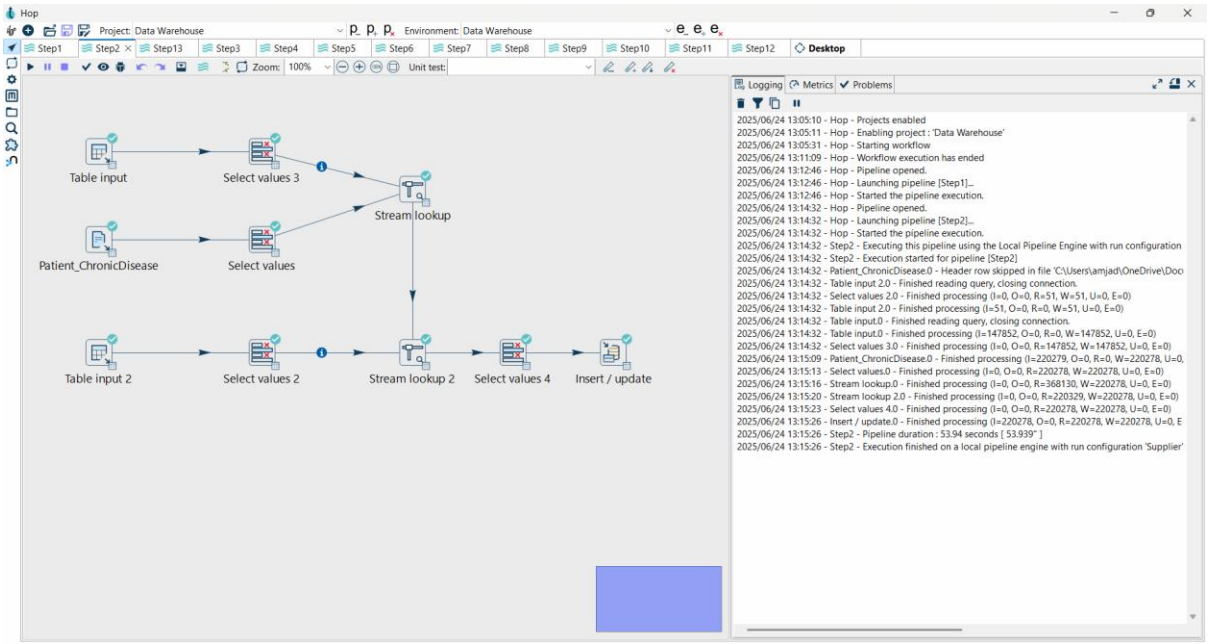
DimSpecialTest

DimTreatment

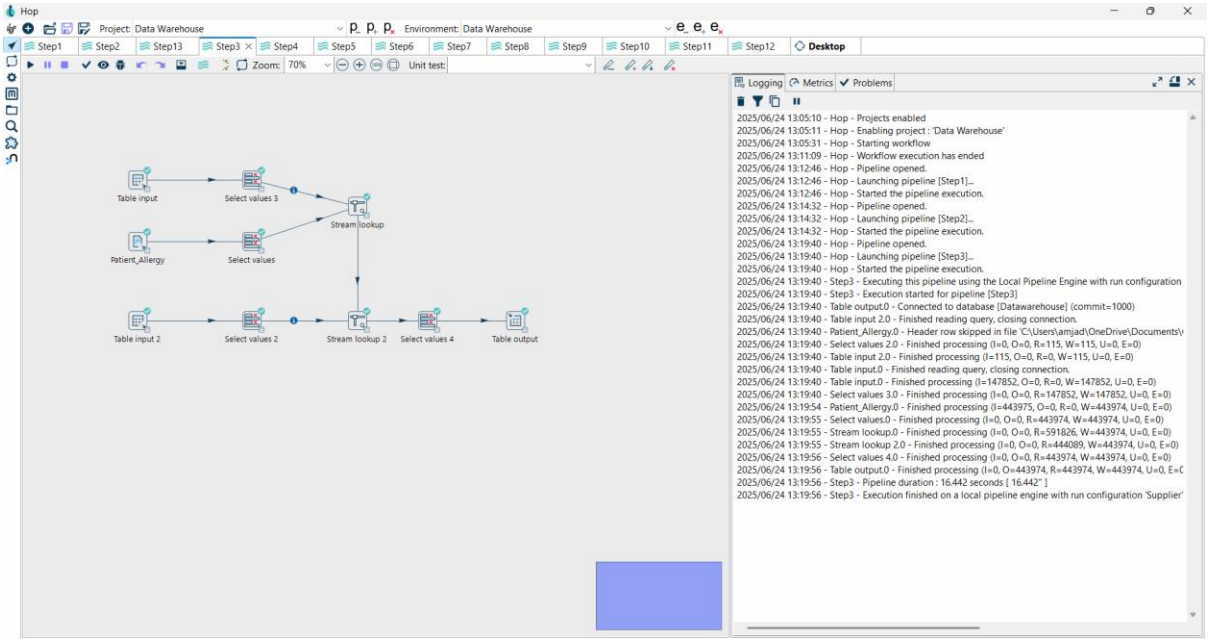
DimAdditionalService



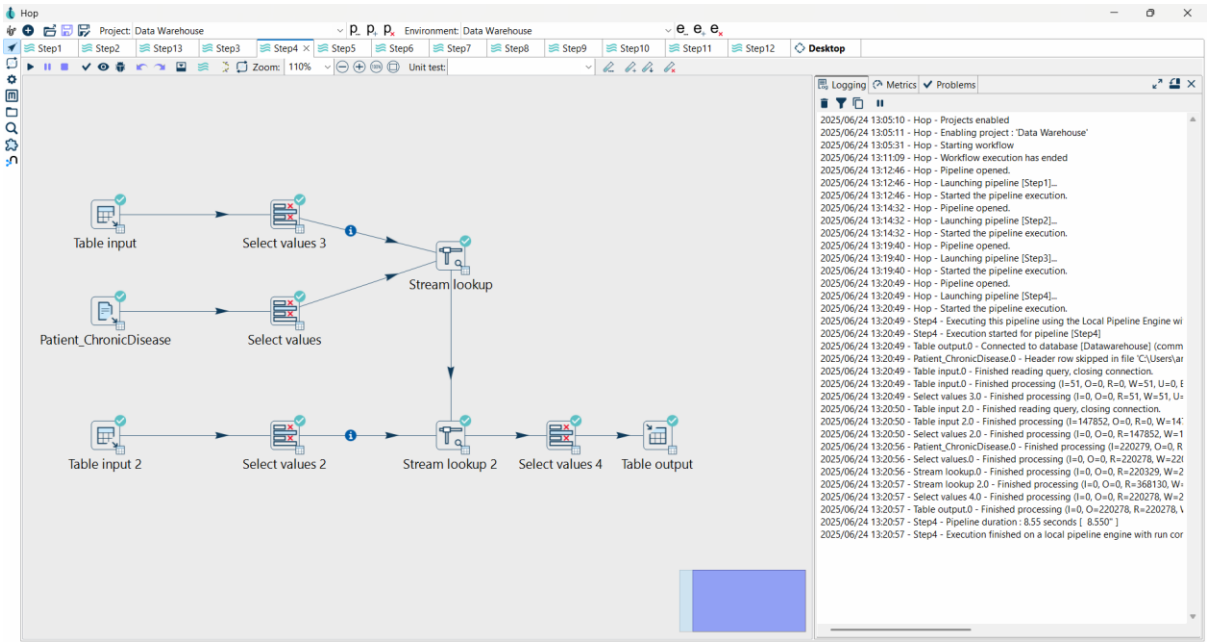
STEP 2 – Patient_ChronicDisease



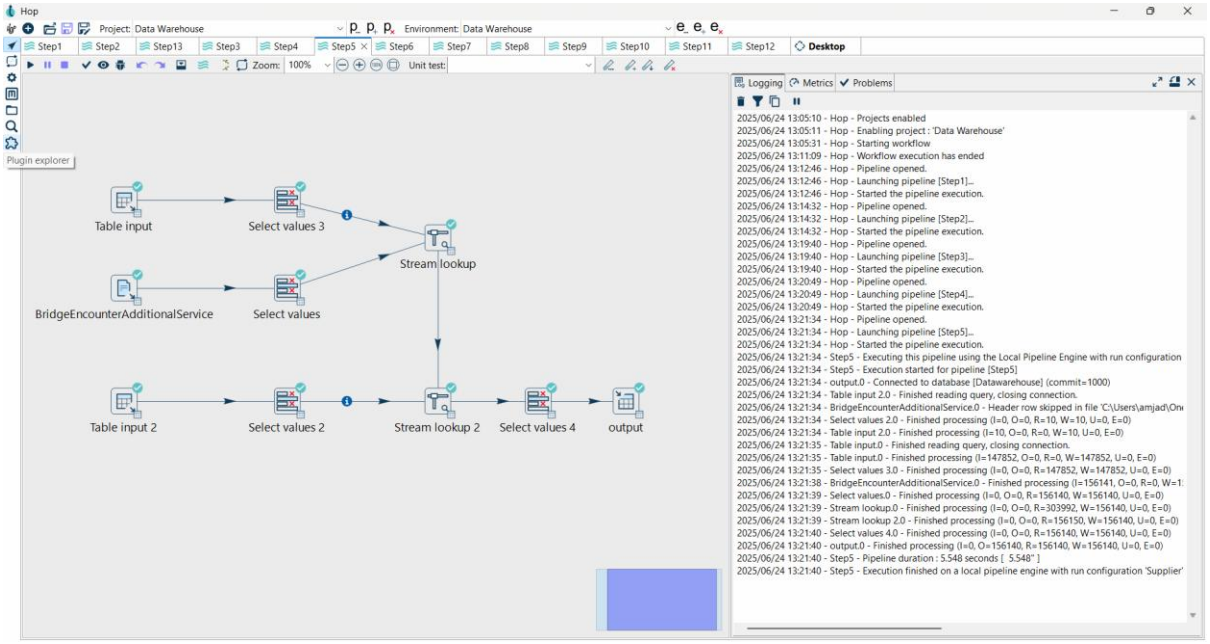
STEP 3 - Patient_Allergy



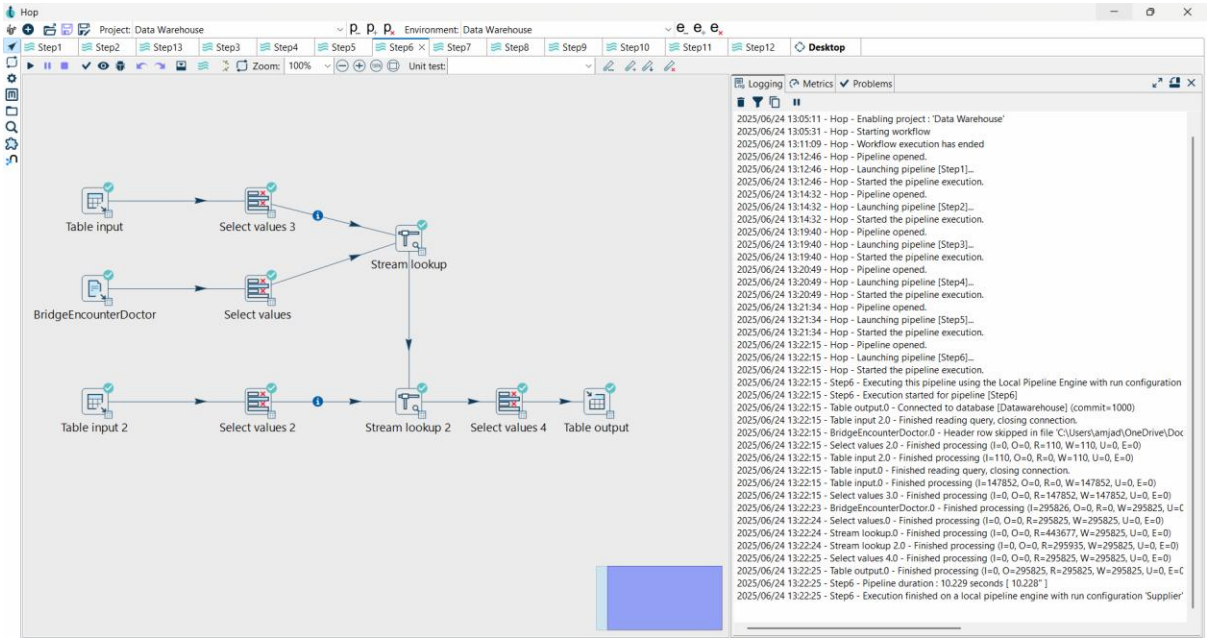
STEP 4 - Patient_AdditionalService



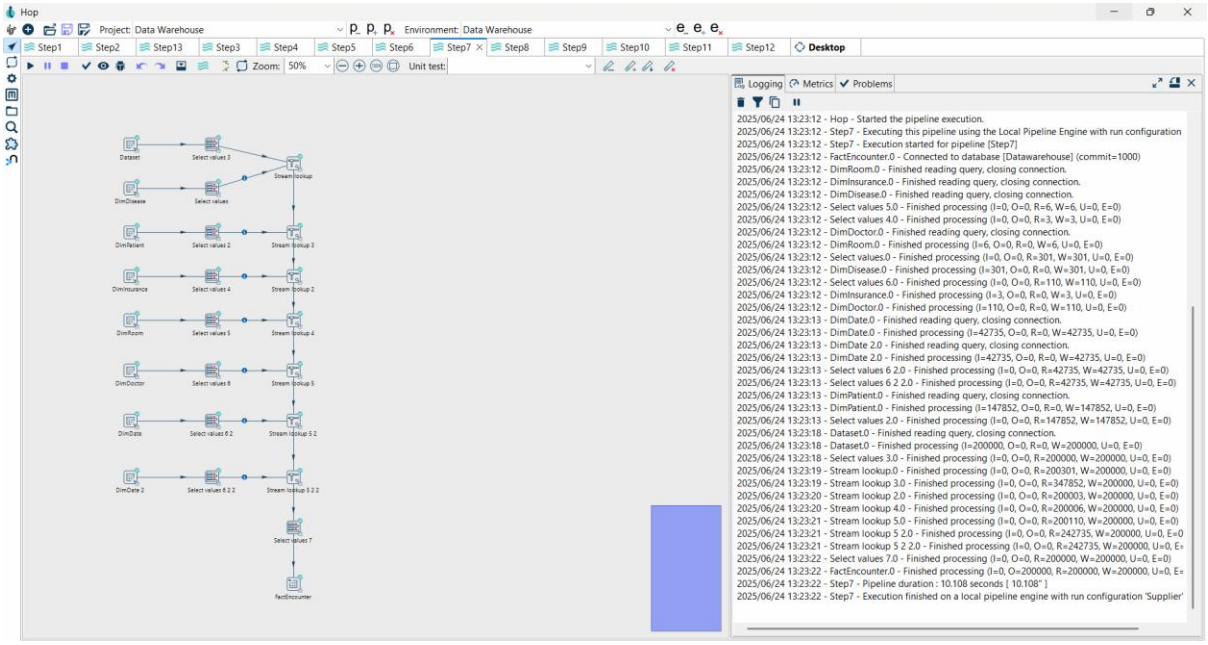
STEP 5 - BridgeEncounterAdditionalService



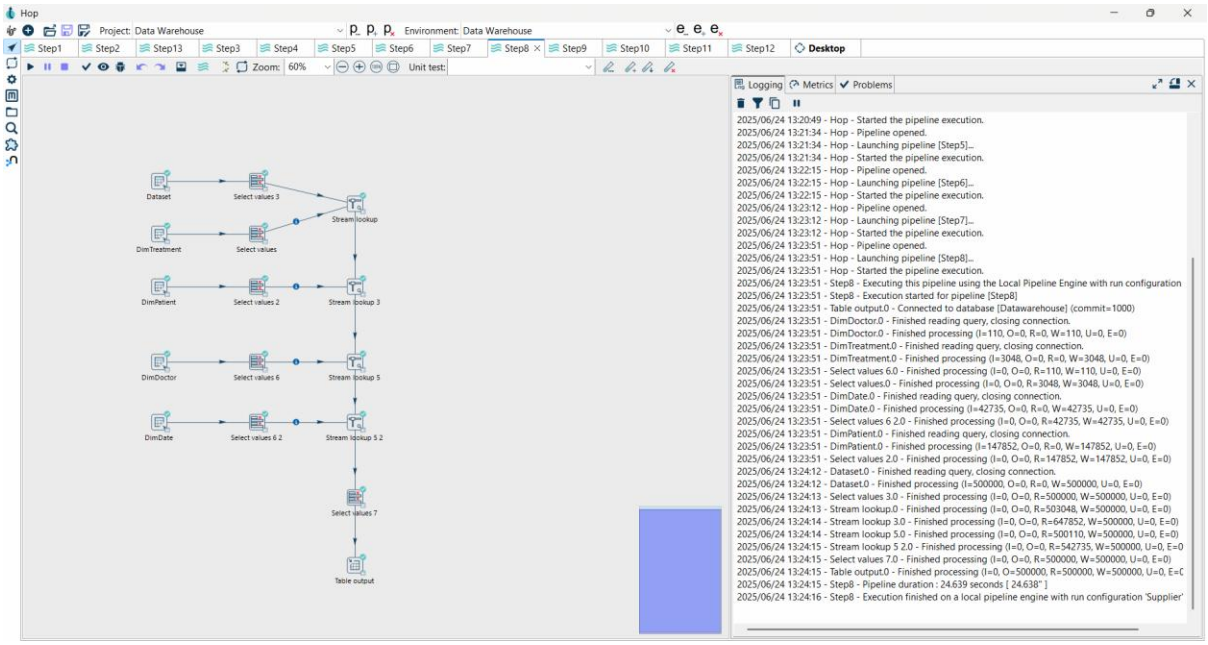
STEP 6 - BridgeEncounterDoctor



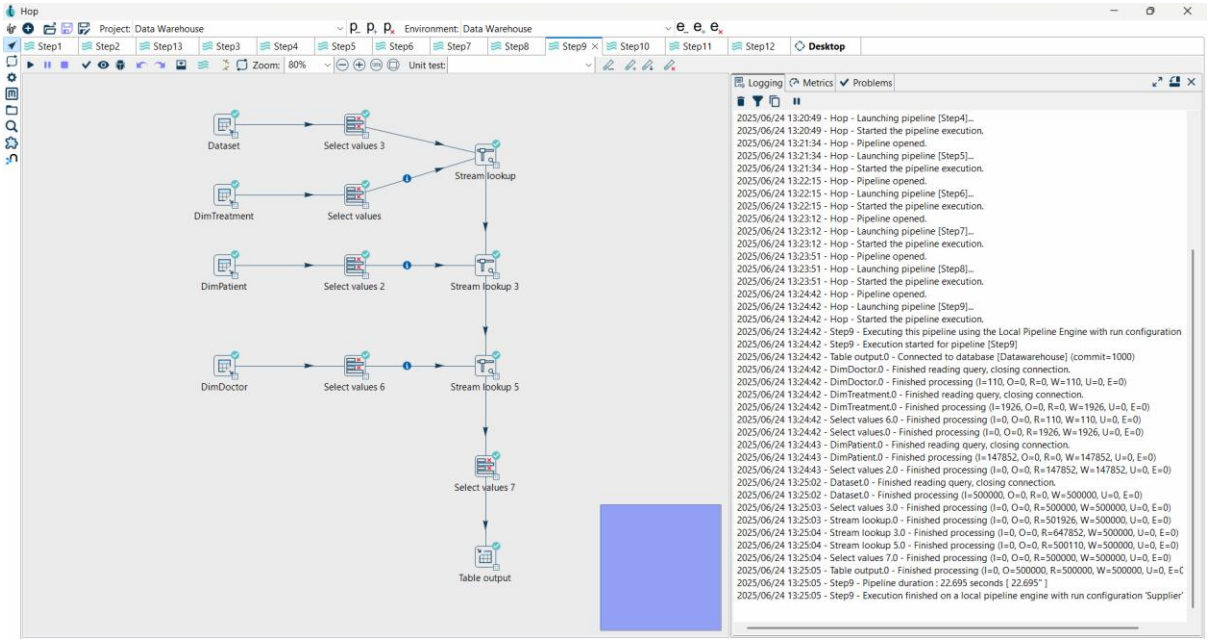
STEP 7 – FactEncounter



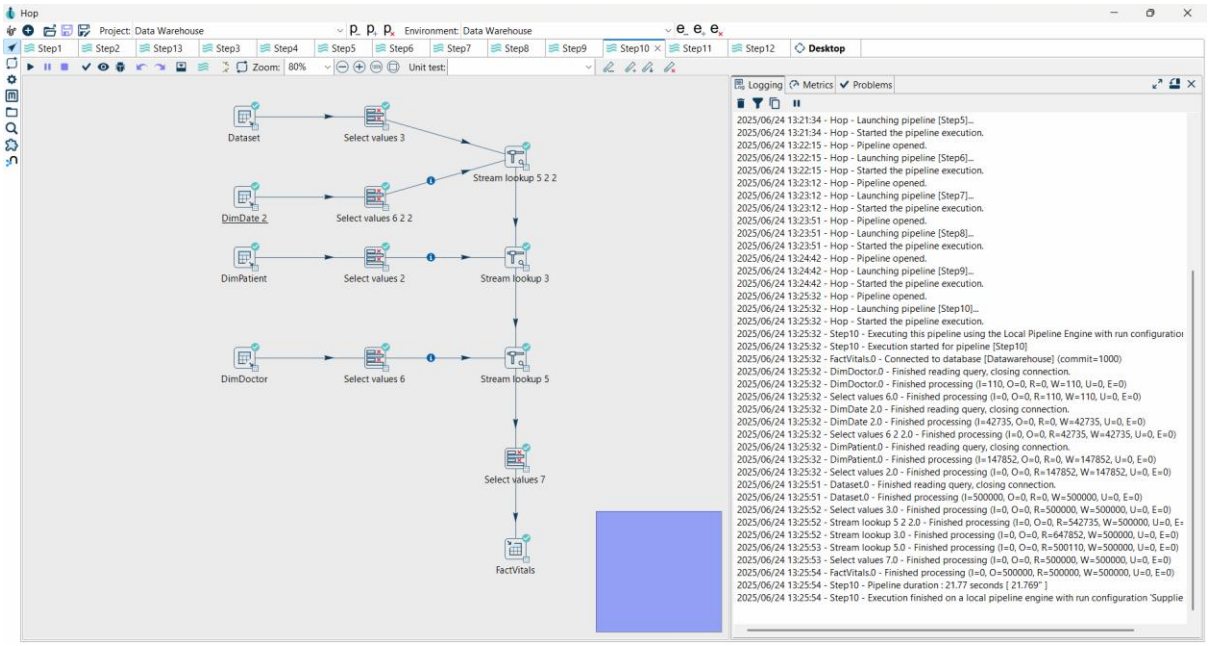
STEP 8 - FactTreatment



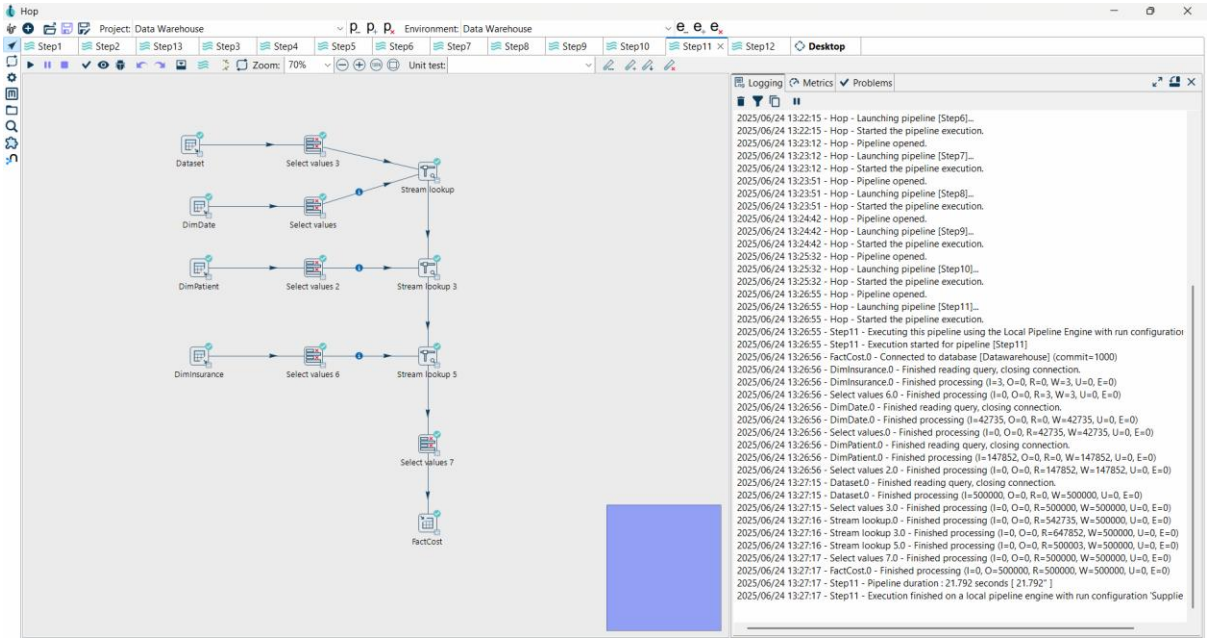
STEP 9 - FactLabTests



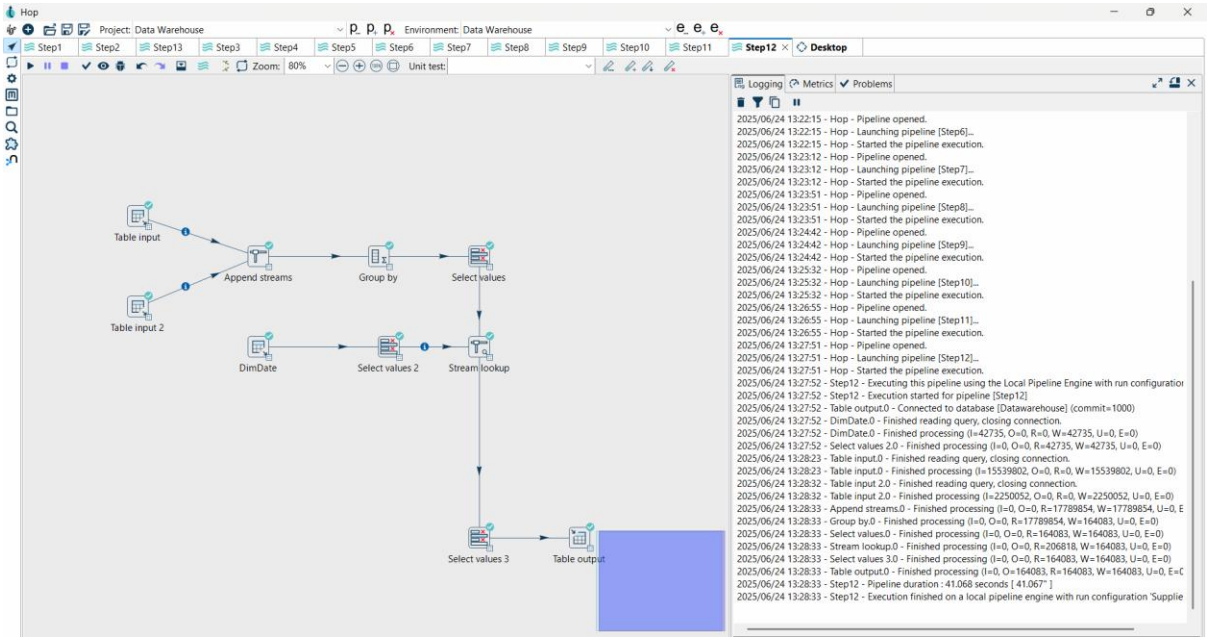
STEP 10 - FactVitals



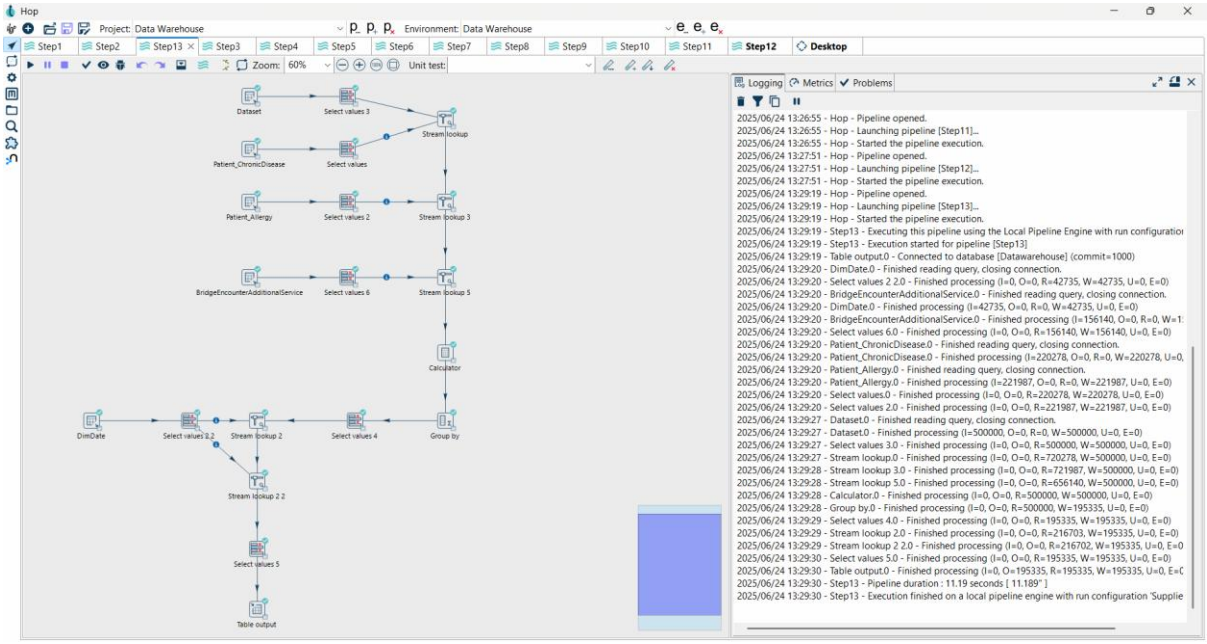
STEP 11- FactCost



STEP 12 - FactProcedures



STEP 13 - FactPatientHealthService



4.3 Source and Target Setup

Apache Hop connected to flat files and SQL Server Mana. Transformation pipeline included joins, nulls handling, formatting, data cleaning and loading into DW normalized tables. Screenshots of these runs are shown in the Appendix.

Chapter 5: Data Visualization Using Power BI

5.1 Data Connection

Power BI wasn't importing the data from the SQL Server database, but was connecting directly to it. Tables were uploaded preserving relationships as in Galaxy Schema.

5.2 Visualizations Created

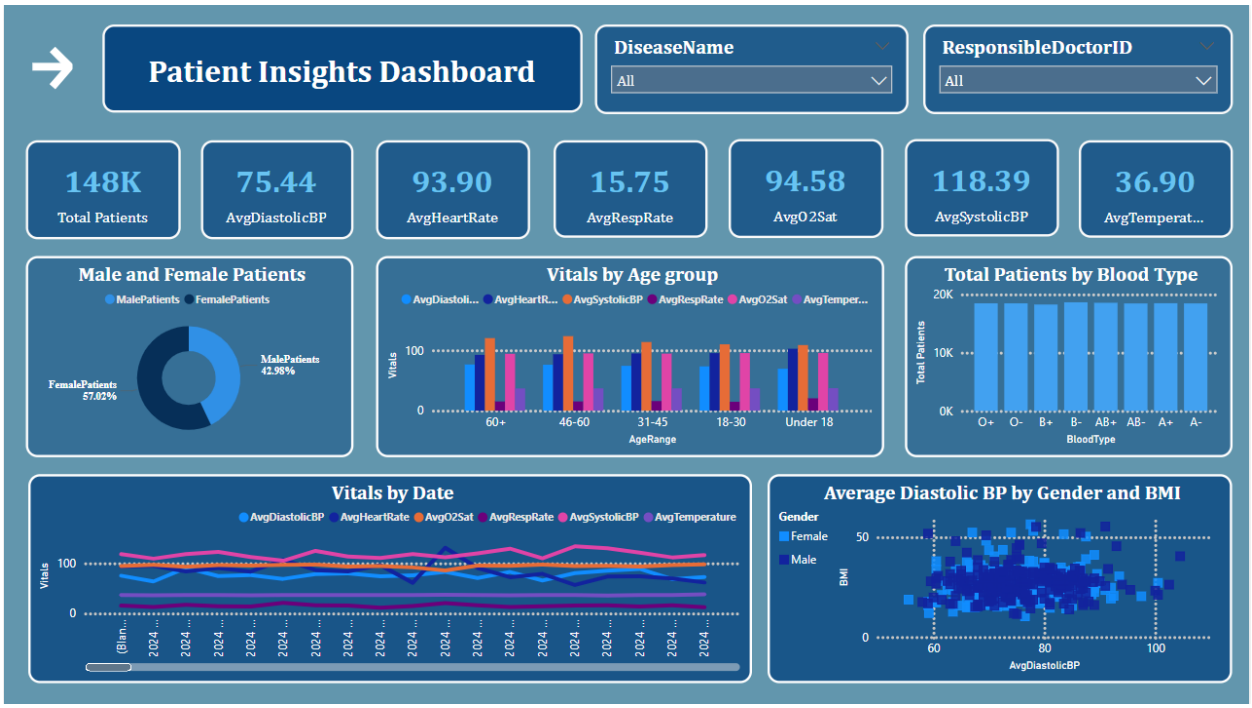
- Slicers are used to filter the dashboard report
- Line and stacked column chart for value comparison and identify trends
- Cards and Gauge used to display values
- Area chart and line chary used to identify trends over time or any other factor
- 100% stacked area chart and 100% stacked bar chart used to show percentages over time
- Clustered bar chart for comparison of factors
- Pie chart and Donut chart for showing percentages as a whole
- Stacked bar chart used to see the relationship between 02 factors where one factor displays in a bar which is split into parts with its subcategories
- The waterfall chart shows an increase and decreases of values.
- Key influences for analyzing the factors which influence any outcomes.
- Decomposition tree used as a step by step breakdown to understand the result

5.3 Dashboard Layout

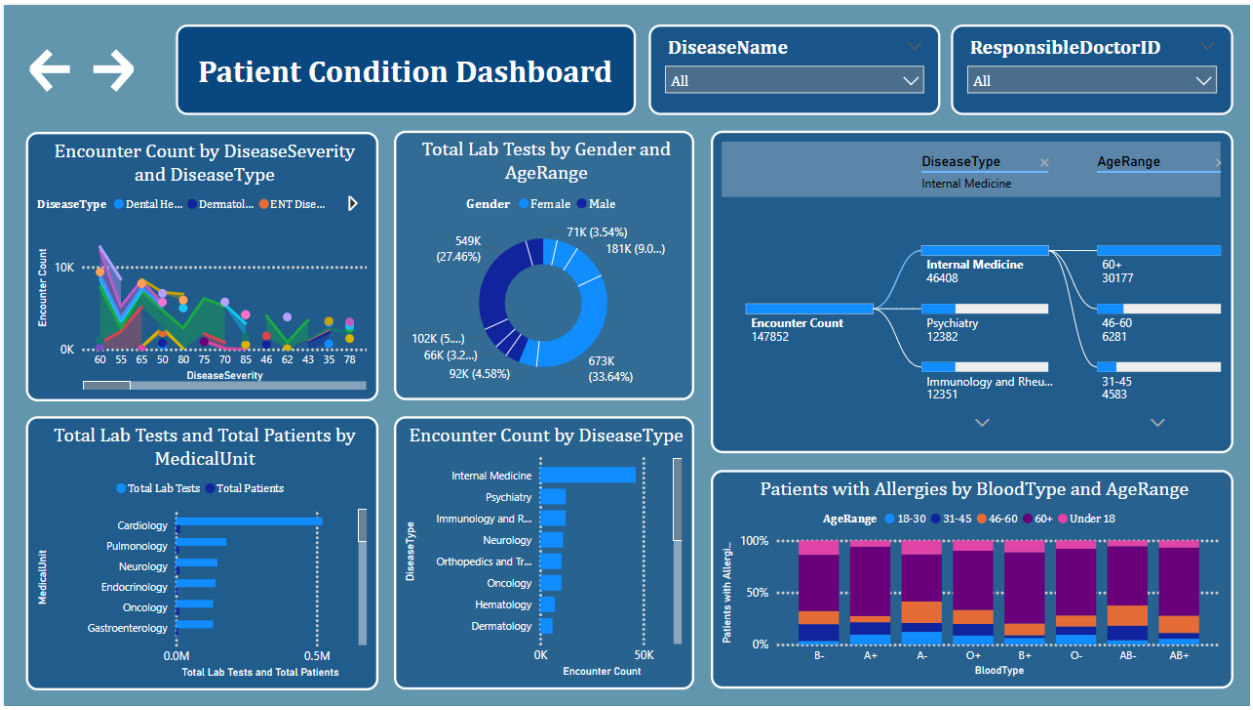
The dashboard included multiple pages:

- Patient Insights
- Patient Condition
- Clinical encounters and diagnostics
- Cost, insurance & Service utilization

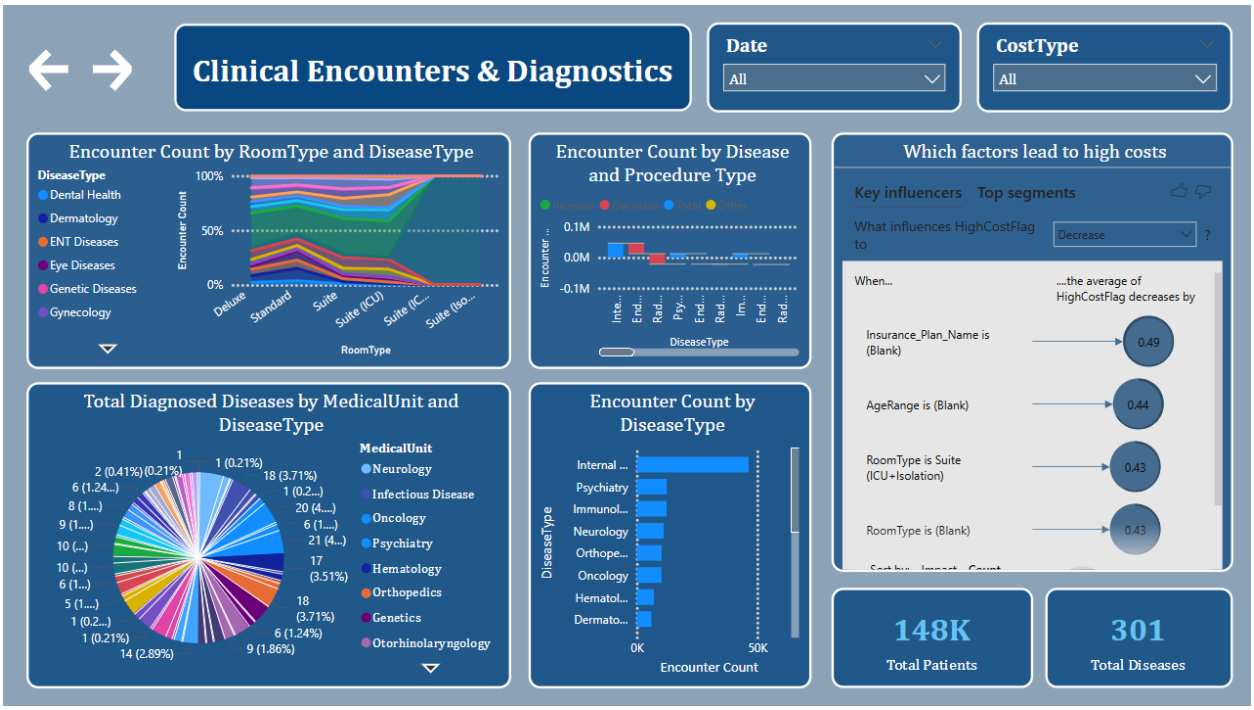
1. Patient Insights



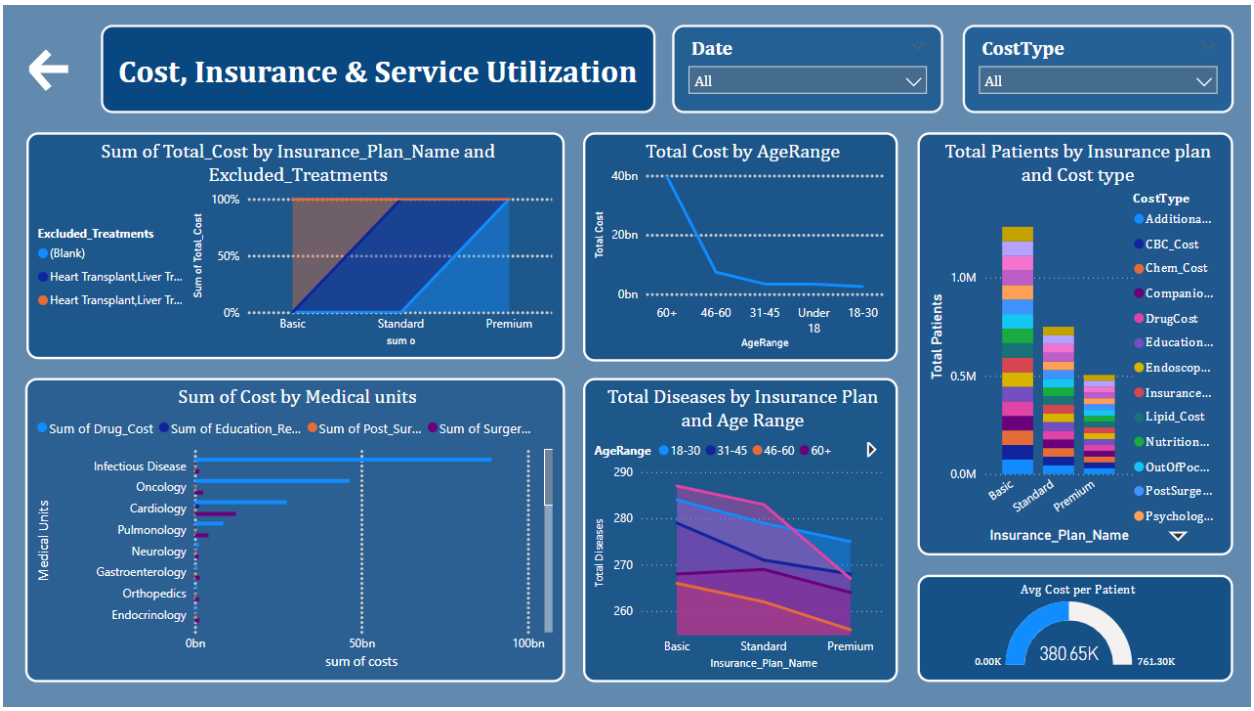
2. Patient Condition



3. Clinical encounters and diagnostics



4. Cost, insurance & Service utilization



5.4 Insights Discovered

Expensive Surgery was Associated with Long Length of Stay.

- Elderly patients had more chronic diseases.
- Some doctors simply saw a higher proportion of sickest patients.
- Testing on labs affected the treatment results.
- Prices varied by season, with the highest prices during high admission periods.

Chapter 6: Findings & Recommendations

6.1 Key Business Insights

- Admissions: Seasonal with peaks in Q1 and Q4.
- Vitals: Low O2 was associated with higher EDRs
- Effect on Treatment: Less follow-up if intervention was early
- Economical: Education and rehab expenditure were quite high although often uncounted.
- Workload: Some doctors, particularly those in chronic care, were overburdened

6.2 Recommendations

- Recognition of complications.
- Invest in chronic disease units.
- Enhance monitoring of costs associated with education and postsurgery.
- Leveraging seasonal patterns to optimize both staffing and inventory.

Chapter 7: Challenges & Solutions

Challenge	Solution
Handling multiple data formats	Used Apache Hop's multi-format connectors
Missing or inconsistent data	Nullable handling and validation logic are in use
Complex joins in bridge tables	Composite keys and intermediary tables were used
ETL Errors - by Data Types	Standardized data before transformation
Dashboard refresh lags	Also improve DAX queries and design of the tables.

Lessons Learned -

- Data cleanup is what makes ETL efficient and normalizing data is key to this.
- Minimize transformation overhead by pre-cleaning data sources.
- Galaxy Schema allows more analytical richness star schema.

Chapter 8: Conclusion

This work has successfully established a robust health care data warehouse with the MedSynora DW data set. By applying a Galaxy Schema, federating a broad range of source systems and modern approaches like Apache Hop or PowerBI, the project provided a scalable solution to support clinical, operational and financial decisions in a hospital. The lessons learned/discovery is that data is king in healthcare management as can be seen from the Power BI dashboard. The modular ETL process is also designed so that next editions for data will be simple to add in, and this keeps the warehouse up to date.

References

- Dataset - Kaggle: https://www.kaggle.com/datasets/mebrar21/medsynora-dw
- Microsoft SQL Server Studio
- Apache Hop Official Documentation
- Power BI Documentation – Microsoft