

**IST 722 Data Warehouse – Project Documentation**

Optimizing ICU Management with Data Warehousing



April 29, 2025

Syracuse University

**Warehouse Wizards**  
***Team Members***  
Bhavesh Purohit  
Simran Ghandhi  
Khushal Modi

Table of Contents

[**IST 722 Data Warehouse – Project Documentation** 0](#_Toc196856392)

[1. Project Overview 2](#_Toc196856393)

[a. Introduction 2](#_Toc196856394)

[b. Project Objectives 2](#_Toc196856395)

[2. Project Plan 3](#_Toc196856396)

[a. Timeline and Milestones 3](#_Toc196856397)

[b. Tasks and Deliverables 3](#_Toc196856398)

[3. Data Overview 4](#_Toc196856399)

[a. Source Table & Description 4](#_Toc196856400)

[b. Data Cleansing & Preparation 5](#_Toc196856401)

[4. Dimensional Modeling 6](#_Toc196856402)

[a. Bus Matrix 6](#_Toc196856403)

[b. Star Schema Overview 7](#_Toc196856404)

[c. Dimension Table Definitions 9](#_Toc196856405)

[d. Fact Table Definitions 9](#_Toc196856406)

[5. Data Warehouse Implementation 10](#_Toc196856407)

[6. Business Intelligence (BI) Implementation 29](#_Toc196856408)

[a. ICU Mortality & Length of Stay Analysis 29](#_Toc196856409)

[b. Medication Usage and Administration Trends 30](#_Toc196856410)

[c. Hospital Resource Utilization & Occupancy Analysis 31](#_Toc196856411)

[7. Challenges and Solutions 32](#_Toc196856412)

[8. Reflection and Next Steps 34](#_Toc196856413)

# 1. Project Overview

## a. Introduction

* Intensive Care Units (ICUs) play a critical role in patient survival but managing ICU resources and predicting patient outcomes remain significant challenges.
* The ICU Data Insights project leverages advanced data warehousing and analytics to extract meaningful patterns from ICU patient records.
* Using the MIMIC-III Clinical Database CareVue subset, this project aims to develop a data-driven approach to enhance clinical decision-making, optimize ICU resource allocation, and improve patient care.

## b. Project Objectives

* To build an integrated, clean, and analytical data warehouse based on ICU operational data.
* To design meaningful business intelligence dashboards for ICU mortality analysis, medication usage trends, and hospital occupancy insights.
* To enable hospital administrators and ICU staff to make data-driven decisions for improving resource utilization and patient outcomes.

# 2. Project Plan

## a. Timeline and Milestones

* **Week 1-2:** Requirements gathering, data profiling, ETL setup.
* **Week 3:** Dimensional modeling (high-level and detailed).
* **Week 4:** Data warehouse build, ETL development.
* **Week 5:** BI dashboard development and validation.
* **Week 6:** Final presentation and documentation.
* Milestone 1: Bus Matrix and High-Level Design
* Milestone 2: Detailed Dimensional Modelling and ETL Implementation
* Milestone 3: BI Dashboard Creation and Demonstration

## b. Tasks and Deliverables

* Dimensional model design and validation.
* ETL pipeline scripts and documentation.
* Power BI dashboard with KPIs and filters.

# 3. Data Overview

This project utilizes a curated subset of the MIMIC-III dataset, which contains de-identified health-related data associated with over 40,000 ICU patients. The data has been profiled, cleaned, and modeled to support ICU-specific analytical goals such as mortality analysis, medication trends, and occupancy monitoring.

## a. Source Table & Description

|  |  |
| --- | --- |
| ADMISSIONS.csv | It contains hospital admissions records, including admit/discharge timestamps, admission types, insurance, and mortality flags. |
| ICUSTAYS.csv | Captures detailed information on each ICU stay, including timestamps, care units, and length of stay. |
| PATIENTS.csv | Includes demographic information such as gender, date of birth, and death dates. |
| TRANSFERS.csv | Records the movement of patients between care units or wards during their hospital stay. |
| PRESCRIPTIONS.csv | Contains data on medications prescribed during a patient’s hospital stay, including start/end times, drug names, dosages, and administration routes. |
| D\_ICD\_PROCEDURES.csv | Descriptive metadata for ICD-9 procedure codes used in the hospital. |

## b. Data Cleansing & Preparation

1. Missing Value Handling

* Null values for key fields like startdate, enddate, drug\_name\_generic, and icustay\_id were replaced with default placeholders (e.g., '1900-01-01' or 'N/A') to maintain integrity in joins.
* ICU stay duration (LOS) was recalculated using intime and outtime to ensure consistency.

1. Derived Attributes

* Age at Admission: Derived using dob and intime from ICU data.
* Mortality Flags: Based on hospital\_expire\_flag and expire\_flag to track in-hospital and post-discharge death.
* ICU Readmission Flag: Computed based on multiple ICU stay entries for a single hadm\_id.

1. Data Validation Checks

* Null count checks on all key fields (e.g., hadm\_id, subject\_id, drug, intime, outtime).
* De-duplication of records for patients and prescriptions.
* Filtering records with missing ICU timestamps.

1. Data Modeling Layers

* Raw Layer (mimic\_dw.raw): Stores raw ingested data from CSVs without transformation.
* Staging Layer stage (views): Performs type casting, initial cleansing, and derivation logic.
* Modeled/Analytical Layer (mimic\_dw.dw): Contains conformed dimensions and fact tables structured in a star schema optimized for BI analysis.

# 4. Dimensional Modeling

The dimensional model for the ICU Data Warehouse is designed using the star schema architecture to enable efficient querying, fast aggregation, and intuitive business reporting through Power BI.

The model supports three core business processes:

1. ICU Mortality and LOS Analysis
2. Medication Usage and Administration Trends
3. Hospital Resource Utilization and Occupancy Analysis

## a. Bus Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Business Process Case** | **ICU Stay Facts** | **Admissions** | **Patients** | **Medications** | **Time** |
| ICU Utilization Analysis | ✅ | ✅ | ✅ | ❌ | ✅ |
| Mortality & Outcome Tracking | ✅ | ✅ | ✅ | ❌ | ✅ |
| Medication Usage Monitoring | ❌ | ✅ | ✅ | ✅ | ✅ |
| Readmission Rate Analysis | ✅ | ✅ | ✅ | ❌ | ✅ |
| Demographics & Risk Breakdown | ✅ | ✅ | ✅ | ❌ | ✅ |
| Drug Prescribing Behavior | ❌ | ✅ | ✅ | ✅ | ✅ |
| Care Unit Transfer Mapping | ✅ | ✅ | ✅ | ❌ | ✅ |

## b. Star Schema Overview

The schema is organized around three main fact tables, each linked to conformed dimensions:

**Fact Tables:**

fact\_icu\_stay

fact\_medication\_admin

fact\_transfer

**Dimension Tables:**

dim\_patient

dim\_admission

dim\_procedure

dim\_service

A screenshot of a computer screen

AI-generated content may be incorrect.dim\_caregiver (in extended modeling)

A computer screen shot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

c. Dimension Table Definitions

|  |  |  |  |
| --- | --- | --- | --- |
| **Dimension Name** | **Primary Key** | **Description** | **Key Attributes** |
| **dim\_patient** | subject\_id | Stores patient-level demographic and vital status data. | gender, dob, age, expire\_flag |
| **dim\_admission** | hadm\_id | Describes each hospital admission. | admission\_type, admission\_location, discharge\_location, insurance, ethnicity, diagnosis |
| **dim\_procedure** | icd9\_code | Metadata about ICD-9 procedure codes. | short\_title, long\_title |
| **dim\_service** | Composite: subject\_id, hadm\_id | Tracks service transitions during the hospital stay. | transfertime, curr\_service |
| **dim\_caregiver** | cgid | Details of ICU caregivers and their roles. | label, description |

## d. Fact Table Definitions

|  |  |  |  |
| --- | --- | --- | --- |
| **Fact Table** | **Grain** | **Description** | **Key Measures / Derived Fields** |
| **fact\_icu\_stay** | One row per ICU stay (icustay\_id) | Captures ICU admissions including LOS, admission type, mortality, and patient demographics. | icu\_stay\_hours, icu\_stay\_days, first\_careunit, last\_careunit, expired, died\_in\_hospital, age |
| **fact\_medication\_admin** | One row per drug administration | Contains medication usage records during ICU or hospital stay. | drug, route, startdate, enddate, medication\_duration, icu\_readmission\_flag |
| **fact\_transfer** | One row per transfer event | Tracks movements between care units during ICU/hospitalization. | prev\_careunit, curr\_careunit, duration\_hours, transfer\_start, transfer\_end |

# 5. Data Warehouse Implementation

**a. Technical Architecture**

* Snowflake as the cloud data warehouse platform.
* Raw, staging, and star schema layers implemented.

**b. Data Pipeline Design (ETL/ELT)**

* **Extract:** Data extracted from CSV using Snowflake stages.
* **Transform:** Data cleaned and enriched using SQL transformations.
* **Load:** Data loaded into fact and dimension tables.

CODE

* 1. **ICU Mortality and LOS Analysis**

-- Create project database

CREATE OR REPLACE DATABASE mimic\_project;

CREATE OR REPLACE FILE FORMAT mimic\_csv

TYPE = 'CSV'

FIELD\_OPTIONALLY\_ENCLOSED\_BY = '"'

SKIP\_HEADER = 1

NULL\_IF = ('\\N', 'NULL', '');

CREATE OR REPLACE STAGE mimic\_stage

FILE\_FORMAT = mimic\_csv;

-- Create 3 schemas for ELT layering

CREATE OR REPLACE SCHEMA mimic\_project.raw;

CREATE OR REPLACE SCHEMA mimic\_project.stage;

CREATE OR REPLACE SCHEMA mimic\_project.modeled;

USE SCHEMA mimic\_project.raw;

CREATE OR REPLACE TABLE icustays (

row\_id INT,

subject\_id INT,

hadm\_id INT,

icustay\_id INT,

dbsource STRING,

first\_careunit STRING,

last\_careunit STRING,

first\_wardid INT,

last\_wardid INT,

intime TIMESTAMP,

outtime TIMESTAMP,

los FLOAT

);

CREATE OR REPLACE TABLE transfers (

row\_id INT,

subject\_id INT,

hadm\_id INT,

icustay\_id FLOAT,

dbsource STRING,

eventtype STRING,

prev\_careunit STRING,

curr\_careunit STRING,

prev\_wardid FLOAT,

curr\_wardid FLOAT,

intime TIMESTAMP,

outtime TIMESTAMP,

los FLOAT

);

CREATE OR REPLACE TABLE admissions (

row\_id INT,

subject\_id INT,

hadm\_id INT,

admittime TIMESTAMP,

dischtime TIMESTAMP,

deathtime TIMESTAMP,

admission\_type STRING,

admission\_location STRING,

discharge\_location STRING,

insurance STRING,

language STRING,

religion STRING,

marital\_status STRING,

ethnicity STRING,

edregtime TIMESTAMP,

edouttime TIMESTAMP,

diagnosis STRING,

hospital\_expire\_flag INT,

has\_chartevents\_data INT

);

CREATE OR REPLACE TABLE patients (

row\_id INT,

subject\_id INT,

gender STRING,

dob TIMESTAMP,

dod TIMESTAMP,

dod\_hosp TIMESTAMP,

dod\_ssn TIMESTAMP,

expire\_flag INT

);

CREATE OR REPLACE TABLE services (

row\_id INT,

subject\_id INT,

hadm\_id INT,

transfertime TIMESTAMP,

prev\_service STRING,

curr\_service STRING

);

-- Load ICU stays

COPY INTO raw.icustays

FROM @mimic\_stage/ICUSTAYS.csv

FILE\_FORMAT = mimic\_csv;

-- Load transfers

COPY INTO raw.transfers

FROM @mimic\_stage/TRANSFERS.csv

FILE\_FORMAT = mimic\_csv;

-- Load admissions

COPY INTO raw.admissions

FROM @mimic\_stage/ADMISSIONS.csv

FILE\_FORMAT = mimic\_csv;

-- Load patients

COPY INTO raw.patients

FROM @mimic\_stage/PATIENTS.csv

FILE\_FORMAT = mimic\_csv;

-- Load services

COPY INTO raw.services

FROM @mimic\_stage/SERVICES.csv

FILE\_FORMAT = mimic\_csv;

-- Load ICU stays

COPY INTO raw.icustays

FROM @mimic\_stage/ICUSTAYS.csv

FILE\_FORMAT = mimic\_csv;

-- Load transfers

COPY INTO raw.transfers

FROM @mimic\_stage/TRANSFERS.csv

FILE\_FORMAT = mimic\_csv;

-- Load admissions

COPY INTO raw.admissions

FROM @mimic\_stage/ADMISSIONS.csv

FILE\_FORMAT = mimic\_csv;

-- Load patients

COPY INTO raw.patients

FROM @mimic\_stage/PATIENTS.csv

FILE\_FORMAT = mimic\_csv;

-- Load services

COPY INTO raw.services

FROM @mimic\_stage/SERVICES.csv

FILE\_FORMAT = mimic\_csv;

USE SCHEMA mimic\_project.stage;

CREATE OR REPLACE VIEW stg\_icustays AS

SELECT

icustay\_id,

subject\_id,

hadm\_id,

first\_careunit,

last\_careunit,

CAST(intime AS TIMESTAMP) AS intime,

CAST(outtime AS TIMESTAMP) AS outtime,

COALESCE(los, DATEDIFF('minute', intime, outtime) / 1440.0) AS los\_days

FROM raw.icustays

WHERE outtime IS NOT NULL;

CREATE OR REPLACE VIEW stg\_transfers AS

SELECT

subject\_id,

hadm\_id,

icustay\_id,

eventtype,

prev\_careunit,

curr\_careunit,

prev\_wardid,

curr\_wardid,

CAST(intime AS TIMESTAMP) AS intime,

CAST(outtime AS TIMESTAMP) AS outtime,

COALESCE(los, DATEDIFF('minute', intime, outtime) / 1440.0) AS los\_days

FROM raw.transfers

WHERE intime IS NOT NULL;

CREATE OR REPLACE VIEW stg\_admissions AS

SELECT

hadm\_id,

subject\_id,

admission\_type,

admission\_location,

discharge\_location,

insurance,

ethnicity,

diagnosis,

hospital\_expire\_flag,

CAST(admittime AS TIMESTAMP) AS admittime,

CAST(dischtime AS TIMESTAMP) AS dischtime,

CAST(deathtime AS TIMESTAMP) AS deathtime

FROM raw.admissions;

CREATE OR REPLACE VIEW stg\_patients AS

SELECT

subject\_id,

gender,

CAST(dob AS TIMESTAMP) AS dob,

CAST(dod AS TIMESTAMP) AS dod,

expire\_flag

FROM raw.patients;

CREATE OR REPLACE VIEW stg\_services AS

SELECT

subject\_id,

hadm\_id,

CAST(transfertime AS TIMESTAMP) AS transfertime,

curr\_service

FROM raw.services;

USE SCHEMA mimic\_project.modeled;

CREATE OR REPLACE VIEW fact\_icu\_stays AS

SELECT

icu.icustay\_id,

icu.subject\_id,

icu.hadm\_id,

icu.first\_careunit,

icu.last\_careunit,

icu.intime,

icu.outtime,

icu.los\_days,

adm.admission\_type,

pat.gender,

DATEDIFF('year', pat.dob, icu.intime) AS age,

adm.hospital\_expire\_flag AS died\_in\_hospital,

pat.expire\_flag AS expired

FROM stage.stg\_icustays icu

LEFT JOIN stage.stg\_admissions adm ON icu.hadm\_id = adm.hadm\_id

LEFT JOIN stage.stg\_patients pat ON icu.subject\_id = pat.subject\_id;

CREATE OR REPLACE VIEW fact\_transfers AS

SELECT

subject\_id,

hadm\_id,

icustay\_id,

eventtype,

prev\_careunit,

curr\_careunit,

intime,

outtime,

los\_days

FROM stage.stg\_transfers

WHERE prev\_careunit IS NOT NULL AND curr\_careunit IS NOT NULL;

CREATE OR REPLACE VIEW dim\_patients AS

SELECT

subject\_id,

gender,

dob,

dod,

expire\_flag

FROM stage.stg\_patients;

CREATE OR REPLACE VIEW dim\_admissions AS

SELECT

hadm\_id,

subject\_id,

admission\_type,

admission\_location,

discharge\_location,

insurance,

ethnicity,

diagnosis,

admittime,

dischtime,

deathtime

FROM stage.stg\_admissions;

CREATE OR REPLACE VIEW dim\_services AS

SELECT

subject\_id,

hadm\_id,

transfertime,

curr\_service

FROM stage.stg\_services;

* 1. **Medication Usage and Administration Trends**

CREATE OR REPLACE DATABASE mimic\_dw;

CREATE OR REPLACE SCHEMA mimic\_dw.raw;

CREATE OR REPLACE SCHEMA mimic\_dw.dw;

-- Create or use your virtual warehouse

USE WAREHOUSE compute\_wh;

drop table mimic\_dw.raw.prescriptions

----Create table for PRESCRIPTIONS

CREATE OR REPLACE TABLE mimic\_dw.raw.prescriptions (

row\_id INT,

subject\_id INT,

hadm\_id INT,

icustay\_id INT,

startdate TIMESTAMP\_NTZ,

enddate TIMESTAMP\_NTZ,

drug\_type STRING,

drug STRING,

drug\_name\_poe STRING,

drug\_name\_generic STRING,

formulary\_drug\_cd STRING,

gsn STRING,

ndc STRING,

prod\_strength STRING,

dose\_val\_rx STRING,

dose\_unit\_rx STRING,

form\_val\_disp STRING,

form\_unit\_disp STRING,

route STRING

);

SELECT \* FROM mimic\_dw.raw.prescriptions LIMIT 500;

UPDATE mimic\_dw.raw.prescriptions

SET

drug\_name\_poe = COALESCE(drug\_name\_poe, 'N/A'),

drug\_name\_generic = COALESCE(drug\_name\_generic, 'N/A')

WHERE

drug\_name\_poe IS NULL OR drug\_name\_generic IS NULL;

UPDATE mimic\_dw.raw.prescriptions

SET

icustay\_id = -1

WHERE

icustay\_id IS NULL;

SELECT

COUNT(\*) AS total\_rows,

COUNT\_IF(row\_id IS NULL) AS null\_row\_id,

COUNT\_IF(subject\_id IS NULL) AS null\_subject\_id,

COUNT\_IF(hadm\_id IS NULL) AS null\_hadm\_id,

COUNT\_IF(icustay\_id IS NULL) AS null\_icustay\_id,

COUNT\_IF(startdate IS NULL) AS null\_startdate,

COUNT\_IF(enddate IS NULL) AS null\_enddate,

COUNT\_IF(drug\_type IS NULL) AS null\_drug\_type,

COUNT\_IF(drug IS NULL) AS null\_drug,

COUNT\_IF(drug\_name\_poe IS NULL) AS null\_drug\_name\_poe,

COUNT\_IF(drug\_name\_generic IS NULL) AS null\_drug\_name\_generic,

COUNT\_IF(formulary\_drug\_cd IS NULL) AS null\_formulary\_drug\_cd,

COUNT\_IF(gsn IS NULL) AS null\_gsn,

COUNT\_IF(ndc IS NULL) AS null\_ndc,

COUNT\_IF(prod\_strength IS NULL) AS null\_prod\_strength,

COUNT\_IF(dose\_val\_rx IS NULL) AS null\_dose\_val\_rx,

COUNT\_IF(dose\_unit\_rx IS NULL) AS null\_dose\_unit\_rx,

COUNT\_IF(form\_val\_disp IS NULL) AS null\_form\_val\_disp,

COUNT\_IF(form\_unit\_disp IS NULL) AS null\_form\_unit\_disp,

COUNT\_IF(route IS NULL) AS null\_route

FROM

mimic\_dw.raw.prescriptions;

UPDATE mimic\_dw.raw.prescriptions

SET

startdate = '1900-01-01 00:00:00'

WHERE startdate IS NULL;

UPDATE mimic\_dw.raw.prescriptions

SET

enddate = '1900-01-01 00:00:00'

WHERE enddate IS NULL;

-----Create table for ADMISSIONS

CREATE OR REPLACE TABLE mimic\_dw.raw.admissions (

row\_id INT,

subject\_id INT,

hadm\_id INT,

admittime TIMESTAMP\_NTZ,

dischtime TIMESTAMP\_NTZ,

deathtime TIMESTAMP\_NTZ,

admission\_type STRING,

admission\_location STRING,

discharge\_location STRING,

insurance STRING,

language STRING,

religion STRING,

marital\_status STRING,

ethnicity STRING,

edregtime TIMESTAMP\_NTZ,

edouttime TIMESTAMP\_NTZ,

diagnosis STRING,

hospital\_expire\_flag INT,

has\_chartevents\_data INT

);

SELECT \* FROM mimic\_dw.raw.admissions LIMIT 500;

UPDATE mimic\_dw.raw.admissions

SET

deathtime = '1900-01-01 00:00:00'

WHERE deathtime IS NULL;

-----Create a table for Patients

CREATE OR REPLACE TABLE mimic\_dw.raw.patients (

row\_id INT,

subject\_id INT,

gender STRING,

dob TIMESTAMP\_NTZ,

dod TIMESTAMP\_NTZ,

dod\_hosp TIMESTAMP\_NTZ,

dod\_ssn TIMESTAMP\_NTZ,

expire\_flag INT

);

SELECT \* FROM mimic\_dw.raw.patients LIMIT 500;

SELECT

COUNT(\*) AS total\_rows,

COUNT\_IF(row\_id IS NULL) AS null\_row\_id,

COUNT\_IF(subject\_id IS NULL) AS null\_subject\_id,

COUNT\_IF(gender IS NULL) AS null\_gender,

COUNT\_IF(dob IS NULL) AS null\_dob,

COUNT\_IF(dod IS NULL) AS null\_dod,

COUNT\_IF(dod\_hosp IS NULL) AS null\_dod\_hosp,

COUNT\_IF(dod\_ssn IS NULL) AS null\_dod\_ssn,

COUNT\_IF(expire\_flag IS NULL) AS null\_expire\_flag

FROM

mimic\_dw.raw.patients;

-----Create table for ICUStays

CREATE OR REPLACE TABLE mimic\_dw.raw.icustays (

row\_id INT,

subject\_id INT,

hadm\_id INT,

icustay\_id INT,

dbsource STRING,

first\_careunit STRING,

last\_careunit STRING,

first\_wardid INT,

last\_wardid INT,

intime TIMESTAMP\_NTZ,

outtime TIMESTAMP\_NTZ,

los FLOAT

);

Select \* from mimic\_dw.raw.icustays limit 500;

SELECT

COUNT(\*) AS total\_rows,

COUNT\_IF(row\_id IS NULL) AS null\_row\_id,

COUNT\_IF(subject\_id IS NULL) AS null\_subject\_id,

COUNT\_IF(hadm\_id IS NULL) AS null\_hadm\_id,

COUNT\_IF(icustay\_id IS NULL) AS null\_icustay\_id,

COUNT\_IF(dbsource IS NULL) AS null\_dbsource,

COUNT\_IF(first\_careunit IS NULL) AS null\_first\_careunit,

COUNT\_IF(last\_careunit IS NULL) AS null\_last\_careunit,

COUNT\_IF(first\_wardid IS NULL) AS null\_first\_wardid,

COUNT\_IF(last\_wardid IS NULL) AS null\_last\_wardid,

COUNT\_IF(intime IS NULL) AS null\_intime,

COUNT\_IF(outtime IS NULL) AS null\_outtime,

COUNT\_IF(los IS NULL) AS null\_los

FROM

mimic\_dw.raw.icustays;

-----Create table for Caregivers

CREATE OR REPLACE TABLE mimic\_dw.raw.caregivers (

row\_id INT,

cgid INT,

label STRING,

description STRING

);

Select \* from mimic\_dw.raw.caregivers limit 500;

SELECT

COUNT(\*) AS total\_rows,

COUNT\_IF(row\_id IS NULL) AS null\_row\_id,

COUNT\_IF(cgid IS NULL) AS null\_cgid,

COUNT\_IF(label IS NULL) AS null\_label,

COUNT\_IF(description IS NULL) AS null\_description

FROM

mimic\_dw.raw.caregivers;

----Fact Table fact\_medication\_dimension

CREATE OR REPLACE TABLE mimic\_dw.dw.fact\_medication\_admin AS

SELECT

p.subject\_id,

p.hadm\_id,

COALESCE(p.icustay\_id::STRING, 'N/A') AS icustay\_id, -- handle nulls

p.drug,

p.startdate,

p.enddate,

DATEDIFF(DAY, p.startdate, p.enddate) AS medication\_duration,

p.route,

p.dose\_val\_rx,

p.ndc,

p.gsn,

a.admission\_type,

-- ICU Readmission: more than 1 icustay for the same hadm\_id = 1

CASE

WHEN ic.count\_stays > 1 THEN 1

ELSE 0

END AS icu\_readmission\_flag

FROM mimic\_dw.raw.prescriptions p

LEFT JOIN mimic\_dw.raw.admissions a

ON p.hadm\_id = a.hadm\_id

LEFT JOIN (

SELECT hadm\_id, COUNT(\*) AS count\_stays

FROM mimic\_dw.raw.icustays

GROUP BY hadm\_id

) ic

ON p.hadm\_id = ic.hadm\_id;

SELECT \* FROM mimic\_dw.dw.fact\_medication\_admin LIMIT 10;

-- Total medication records

SELECT COUNT(\*) FROM mimic\_dw.dw.fact\_medication\_admin;

-- Check for nulls in important fields

SELECT COUNT(\*) FROM mimic\_dw.dw.fact\_medication\_admin

WHERE subject\_id IS NULL OR hadm\_id IS NULL OR drug IS NULL;

-- Any duplicate rows?

SELECT subject\_id, hadm\_id, drug, COUNT(\*)

FROM mimic\_dw.dw.fact\_medication\_admin

GROUP BY subject\_id, hadm\_id, drug

HAVING COUNT(\*) > 1;

SELECT

f.subject\_id,

d.gender,

f.drug,

f.medication\_duration,

i.first\_careunit,

a.admission\_type

FROM mimic\_dw.dw.fact\_medication\_admin f

LEFT JOIN mimic\_dw.raw.patients d ON f.subject\_id = d.subject\_id

LEFT JOIN mimic\_dw.raw.icustays i ON f.icustay\_id::INT = i.icustay\_id

LEFT JOIN mimic\_dw.raw.admissions a ON f.hadm\_id = a.hadm\_id

LIMIT 20;

* 1. **Hospital Resource Utilization and Occupancy Analysis**

CREATE OR REPLACE DATABASE hospital\_analytics;

-- Create a schema for organizing tables

CREATE OR REPLACE SCHEMA hospital\_analytics.icu\_analysis;

CREATE OR REPLACE FILE FORMAT hospital\_analytics.icu\_analysis.csv\_format

TYPE = 'CSV'

FIELD\_OPTIONALLY\_ENCLOSED\_BY = '"'

SKIP\_HEADER = 1;

CREATE OR REPLACE STAGE hospital\_analytics.icu\_analysis.data\_stage

FILE\_FORMAT = csv\_format;

CREATE OR REPLACE TABLE admissions (

row\_id INT,

subject\_id INT,

hadm\_id INT,

admittime TIMESTAMP,

dischtime TIMESTAMP,

deathtime TIMESTAMP,

admission\_type STRING,

admission\_location STRING,

discharge\_location STRING,

insurance STRING,

language STRING,

religion STRING,

marital\_status STRING,

ethnicity STRING,

edregtime TIMESTAMP,

edouttime TIMESTAMP,

diagnosis STRING,

hospital\_expire\_flag BOOLEAN,

has\_chartevents\_data BOOLEAN

);

CREATE OR REPLACE TABLE icustays (

row\_id INT,

subject\_id INT,

hadm\_id INT,

icustay\_id INT,

dbsource STRING,

first\_careunit STRING,

last\_careunit STRING,

first\_wardid INT,

last\_wardid INT,

intime TIMESTAMP,

outtime TIMESTAMP,

los FLOAT -- Length of Stay in days

);

CREATE OR REPLACE TABLE transfers (

row\_id INT,

subject\_id INT,

hadm\_id INT,

icustay\_id INT,

dbsource STRING,

eventtype STRING,

prev\_careunit STRING,

curr\_careunit STRING,

prev\_wardid INT,

curr\_wardid INT,

intime TIMESTAMP,

outtime TIMESTAMP,

los FLOAT

);

CREATE OR REPLACE TABLE patients (

row\_id INT,

subject\_id INT,

gender STRING,

dob TIMESTAMP,

dod TIMESTAMP,

dod\_hosp TIMESTAMP,

dod\_ssn TIMESTAMP,

expire\_flag BOOLEAN

);

CREATE OR REPLACE TABLE d\_icd\_procedures (

row\_id INT,

icd9\_code STRING,

short\_title STRING,

long\_title STRING

);

COPY INTO admissions

FROM @hospital\_analytics.icu\_analysis.data\_stage/ADMISSIONS.csv

FILE\_FORMAT = (FORMAT\_NAME = csv\_format);

COPY INTO icustays

FROM @hospital\_analytics.icu\_analysis.data\_stage/ICUSTAYS.csv

FILE\_FORMAT = (FORMAT\_NAME = hospital\_analytics.icu\_analysis.csv\_format);

COPY INTO transfers

FROM @hospital\_analytics.icu\_analysis.data\_stage/TRANSFERS.csv

FILE\_FORMAT = (FORMAT\_NAME = hospital\_analytics.icu\_analysis.csv\_format);

LIST @hospital\_analytics.icu\_analysis.data\_stage;

COPY INTO patients

FROM @hospital\_analytics.icu\_analysis.data\_stage/PATIENTS.csv

FILE\_FORMAT = (FORMAT\_NAME = hospital\_analytics.icu\_analysis.csv\_format);

COPY INTO d\_icd\_procedures

FROM @hospital\_analytics.icu\_analysis.data\_stage/D\_ICD\_PROCEDURES.csv

FILE\_FORMAT = (FORMAT\_NAME = hospital\_analytics.icu\_analysis.csv\_format);

-- Sample: Create view with ICU duration

CREATE OR REPLACE VIEW icu\_stays\_cleaned AS

SELECT

\*,

DATEDIFF(HOUR, intime, outtime) AS duration\_hours

FROM icustays

WHERE intime IS NOT NULL AND outtime IS NOT NULL;

CREATE OR REPLACE VIEW vw\_clean\_patients AS

SELECT

subject\_id,

UPPER(gender) AS gender,

dob,

dod,

expire\_flag,

-- derived: age at death or current age

CASE

WHEN dod IS NOT NULL THEN DATEDIFF(YEAR, dob, dod)

ELSE DATEDIFF(YEAR, dob, CURRENT\_DATE())

END AS age

FROM patients

WHERE gender IS NOT NULL AND dob IS NOT NULL;

CREATE OR REPLACE VIEW vw\_clean\_admissions AS

SELECT

subject\_id,

hadm\_id,

admittime,

dischtime,

COALESCE(deathtime, NULL) AS deathtime,

admission\_type,

admission\_location,

discharge\_location,

COALESCE(insurance, 'UNKNOWN') AS insurance,

COALESCE(language, 'UNKNOWN') AS language,

COALESCE(religion, 'UNKNOWN') AS religion,

COALESCE(marital\_status, 'UNKNOWN') AS marital\_status,

COALESCE(ethnicity, 'UNKNOWN') AS ethnicity,

diagnosis,

hospital\_expire\_flag

FROM admissions

WHERE admittime IS NOT NULL AND dischtime IS NOT NULL;

CREATE OR REPLACE VIEW vw\_clean\_icustays AS

SELECT

subject\_id,

hadm\_id,

icustay\_id,

first\_careunit,

last\_careunit,

intime,

outtime,

-- override los to be calculated from timestamps

DATEDIFF(HOUR, intime, outtime) AS icu\_stay\_hours,

ROUND(DATEDIFF(HOUR, intime, outtime)/24.0, 2) AS icu\_stay\_days

FROM icustays

WHERE intime IS NOT NULL AND outtime IS NOT NULL;

CREATE OR REPLACE VIEW vw\_clean\_transfers AS

SELECT

subject\_id,

hadm\_id,

eventtype,

COALESCE(prev\_careunit, 'UNKNOWN') AS prev\_careunit,

COALESCE(curr\_careunit, 'UNKNOWN') AS curr\_careunit,

intime,

outtime,

-- derived: time spent in transfer unit

DATEDIFF(HOUR, intime, outtime) AS duration\_hours

FROM transfers

WHERE intime IS NOT NULL AND outtime IS NOT NULL;

CREATE OR REPLACE VIEW vw\_clean\_procedures AS

SELECT

icd9\_code,

COALESCE(short\_title, 'UNKNOWN') AS short\_title,

COALESCE(long\_title, 'UNKNOWN') AS long\_title

FROM d\_icd\_procedures

WHERE icd9\_code IS NOT NULL;

CREATE OR REPLACE VIEW vw\_patient\_journey AS

SELECT

a.subject\_id,

a.hadm\_id,

a.admittime,

a.dischtime,

i.intime AS icu\_intime,

i.outtime AS icu\_outtime,

t.intime AS transfer\_start,

t.outtime AS transfer\_end

FROM vw\_clean\_admissions a

LEFT JOIN vw\_clean\_icustays i ON a.hadm\_id = i.hadm\_id

LEFT JOIN vw\_clean\_transfers t ON a.hadm\_id = t.hadm\_id;

CREATE OR REPLACE TABLE dim\_patient AS

SELECT DISTINCT

subject\_id,

gender,

dob,

age,

expire\_flag

FROM vw\_clean\_patients;

CREATE OR REPLACE TABLE dim\_admission AS

SELECT DISTINCT

subject\_id,

hadm\_id,

admission\_type,

admission\_location,

discharge\_location,

insurance,

language,

religion,

marital\_status,

ethnicity,

diagnosis,

hospital\_expire\_flag

FROM vw\_clean\_admissions;

CREATE OR REPLACE TABLE dim\_procedure AS

SELECT DISTINCT

icd9\_code,

short\_title,

long\_title

FROM vw\_clean\_procedures;

CREATE OR REPLACE TABLE fact\_icu\_stay AS

SELECT

icustay\_id,

subject\_id,

hadm\_id,

first\_careunit,

last\_careunit,

intime,

outtime,

icu\_stay\_hours,

icu\_stay\_days

FROM vw\_clean\_icustays;

CREATE OR REPLACE TABLE fact\_transfer AS

SELECT

subject\_id,

hadm\_id,

eventtype,

prev\_careunit,

curr\_careunit,

intime AS transfer\_start,

outtime AS transfer\_end,

duration\_hours

FROM vw\_clean\_transfers;

CREATE OR REPLACE TABLE dim\_patient AS

SELECT subject\_id, gender, dob, expire\_flag FROM patients;

SHOW WAREHOUSES;

# 6. Business Intelligence (BI) Implementation

## a. ICU Mortality & Length of Stay Analysis

A screenshot of a data analysis

AI-generated content may be incorrect.

[**Link to the above Dashboard**](https://sumailsyr-my.sharepoint.com/my?id=%2Fpersonal%2Fbppurohi%5Fsyr%5Fedu%2FDocuments%2FBhavesh%20%40%20Syracuse%20University%2FICU%20Mortality%20%26%20Length%20of%20Stay%20Analysis%2Epbix&parent=%2Fpersonal%2Fbppurohi%5Fsyr%5Fedu%2FDocuments%2FBhavesh%20%40%20Syracuse%20University&ga=1)

**Purpose:**

* To analyze ICU patient outcomes in terms of mortality rates and length of stay (LOS).
* To identify trends that can help hospitals optimize ICU care strategies, reduce mortality, and manage patient stays more efficiently.

**Insights Gained:**

* Longer ICU stays can correlate with higher mortality.
* Certain admission types (e.g., emergency admissions) may be linked to longer stays and higher risk.
* Identifying high-risk age groups can help focus preventative efforts and resource allocation.

## b. Medication Usage and Administration Trends

A close-up of a graph

AI-generated content may be incorrect.

[**Link to the above Dashboard**](https://sumailsyr-my.sharepoint.com/:u:/g/personal/bppurohi_syr_edu/ETBT-LLNZpFGpDzLggA0vvIBu3KZ3soZ0b7wyqCwXflWAg?e=1gbWvy)

**Purpose:**

* To visualize patterns in ICU drug usage across different demographics, timeframes, and ICU types.
* To flag unusual or high-risk drug administration patterns.

**Insights Gained:**

* Some ICU types favor specific drug classes more than others.
* Pediatric or geriatric patients might show different medication patterns.
* Seasonal peaks in certain drug usages (like antibiotics during winter) can be spotted.
* Standardization allowed for accurate aggregation and filtering.

## c. Hospital Resource Utilization & Occupancy Analysis

A screenshot of a medical report

AI-generated content may be incorrect.

[**Link to the above Dashboard**](https://sumailsyr-my.sharepoint.com/:u:/g/personal/bppurohi_syr_edu/EbY2npSzlVNAkGpKa_MzxKUBUu9KNdLxYZozT06KpGhPzQ?e=Ujm1bT)

**Purpose:**

* To optimize hospital resource allocation by tracking ICU occupancy, bed usage rates, and resource strain.
* To support better planning and capacity management based on trends.

**Insights Gained:**

* Certain ICUs face consistently high occupancy, indicating a need for expansion or targeted resource planning.
* Weekdays might have different admission patterns compared to weekends.
* Helps predict when surges are likely (e.g., winter flu season leading to higher ICU admissions).
* Bed turnover rates can be tracked to see if patient stays are getting shorter or longer over time.

# 7. Challenges and Solutions

a. Handling Complex and Large-Scale Healthcare Data (MIMIC-III)

Solution: Conducted thorough data exploration to understand table linkages (e.g., PATIENTS ↔ ADMISSIONS ↔ ICUSTAYS).

Focused on necessary fields for ICU analysis to reduce dataset size for warehousing.

b. Standardizing Inconsistent and Dirty Data

Solution: Created standardized transformations:

Unified medication names using DRUG\_NAME\_POE, DRUG\_NAME\_GENERIC, and GSN.

Derived standard date and time formats and extracted components like day, hour, month.

c. Managing file ingestion, transformation, and querying for large CSVs can become slow and error-prone if not architected properly.

Solution: Leveraged Snowflake’s scalable cloud infrastructure:

Used internal stages for efficient file uploads.

Applied COPY INTO commands for ingestion into RAW tables.

Designed modular SQL scripts for STAGE and MODELED transformations to promote reusability and easier troubleshooting.

Followed ELT (Extract, Load, Transform) best practices — avoiding transformation outside the warehouse for speed and traceability.

d. With vast available data, deciding what KPIs to prioritize for ICU management was challenging.

Solution: Conducted a requirements-gathering session (internal discussion) to narrow focus to three critical business processes:

ICU Mortality and LOS

Medication Usage

Hospital Resource Utilization

Created calculated fields and aggregations at the database level (in Snowflake) instead of overloading Power BI with complex DAX calculations.

e. Ensuring Business Relevance and Real-World Applicability

Solution: Mapped dashboards directly to business outcomes:

Mortality and LOS → Clinical quality improvement.

Medication Trends → Safer prescribing practices.

Resource Utilization → Better ICU staffing and planning.

Explained insights in healthcare-friendly language to bridge the gap between technical implementation and clinical interpretation

# 8. Reflection and Next Steps

**REFLECTIONS**

a. Value of Healthcare-Specific Data Understanding

Working with MIMIC-III taught us that domain knowledge is just as critical as technical skills. Understanding clinical context (e.g., ICU readmissions, medication administration) helped us design relevant metrics and visuals.

b. Power of Structured Data Pipelines

Implementing a layered Snowflake architecture (RAW → STAGE → MODELED) proved invaluable. It gave us flexibility, traceability, and ease in debugging data issues without compromising raw data integrity.

c. Importance of Data Standardization

Even minor inconsistencies in drug names or timestamps can cause major analytical errors. Building transformation logic for cleaning and standardizing records was key to enabling trustworthy insights.

**NEXT STEPS**

a. Integrate Real-Time or Near-Real-Time Data

Expand the system to handle near-real-time data streams to support operational decision-making in fast-moving ICU environments.

b. Add Predictive Capabilities

Use historical trends to build machine learning models for predicting ICU readmissions, expected length of stay, or even risk scoring for mortality — laying the foundation for proactive care.

c. Enhance User Accessibility and Training

Develop user guides and training sessions for clinical staff and analysts to maximize adoption and impact of the dashboards.

**BI Dashboard Link:**

Business Process 1: ICU Mortality & Length of Stay Analysis

*LINK:*

Business Process 2: Medication Usage and Administration Trends

*LINK:*

Business Process 3: Hospital Resource Utilization & Occupancy Analysis

*LINK:*