Patient Health Management System

Milestone: SQL Queries

Group: - Data Dynamos
Muskan Sharma
Hemalikaa Thirumavalavan
Mohit Deshpande

(778)869-1739 (Muskan)
(437)217-5392 (Hemalikaa)
(437)662-4519 (Mohit)

sharma.muskan@northeastern.edu
thirumavalavan.h@northeastern.edu
deshpande.mo@northeastern.edu

Percentage of Effort Contributed by Student1: 33.3%

Percentage of Effort Contributed by Student2: 33.3%

Percentage of Effort Contributed by Student3: 33.3%

Signature of Student 1: Muskan Sharma

Signature of Student 2: Hemalikaa Thirumavalavan

Signature of Student 3: Mohit Deshpande

Submission Date: 10/04/2025

```
# Import necessary libraries
import pymysql
from sqlalchemy import create_engine
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
import numpy as np

# Set up matplotlib to display plots inline in the notebook
%matplotlib inline
plt.style.use('ggplot')
sns.set_palette("Set2")
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)

username = "root"
password = "Deshpandem97!"
host = "localhost"
database = "phms"

engine = create_engine(f"mysql+pymysql://{username}:
{password}@{host}/{database}")

tables = pd.read_sql("SHOW TABLES", engine)
print("□ Tables in Database:\n", tables)
```

```
□ Tables in Database:
          Tables_in_phms
0                billing
1              diagnosis
2       diagnosisdetails
3                   epoc
4     healthcareprovider
5      insuranceprovider
6       labresultdetails
7             labresults
8      medicationdetails
9                patient
10  patientdemographics
11     patientencounter
12         prescription
13       symptomdetails
14             symptoms
15          vaccination
16     vitalsigndetails
17           vitalsigns
```

```
patient_df = pd.read_sql("SELECT * FROM patient", engine)
encounter_df = pd.read_sql("SELECT * FROM patientencounter", engine)
```

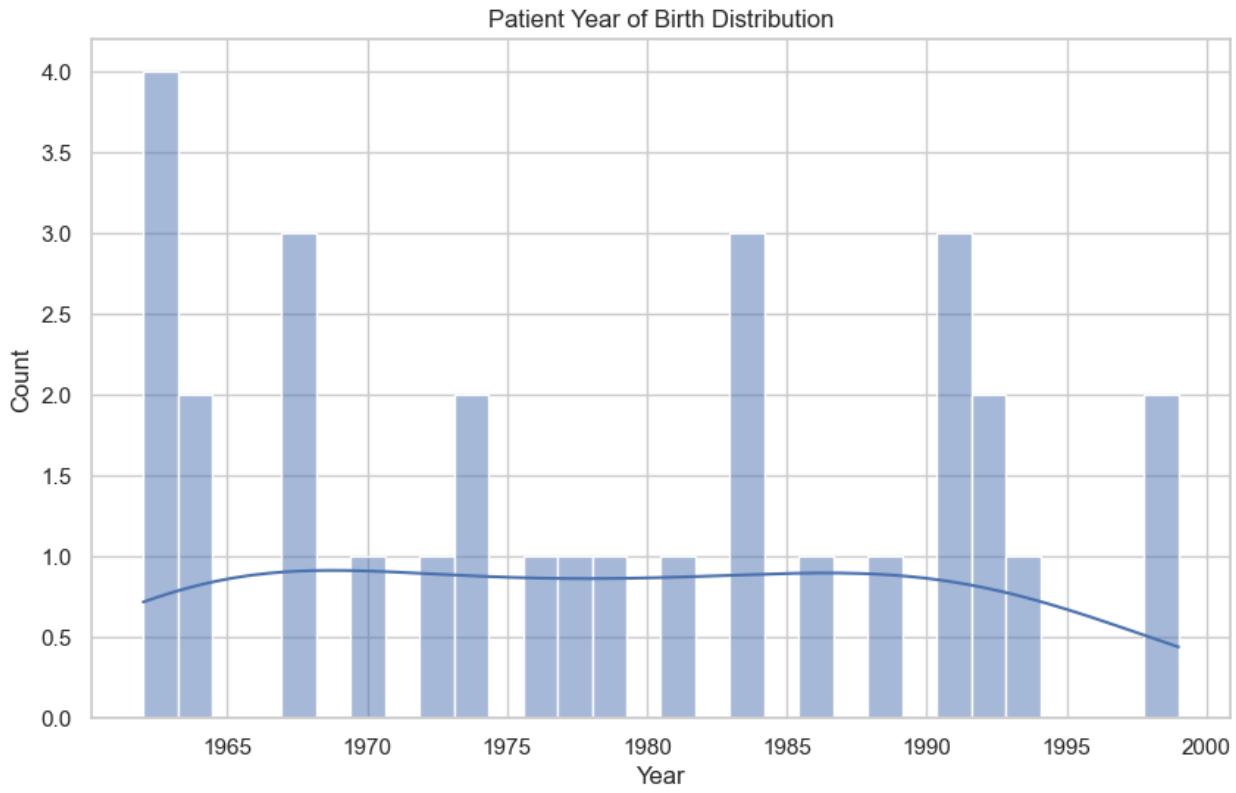
```
billing_df = pd.read_sql("SELECT * FROM billing", engine)
diagnosis_df = pd.read_sql("SELECT * FROM diagnosis", engine)
medications_df = pd.read_sql("SELECT * FROM medicationdetails",
engine)

print(f" Total unique patients: {patient_df['PatID'].nunique()}")
```

```
 Total unique patients: 30
```

```
patient_df['YearOfBirth'] = pd.to_datetime(patient_df['DoB'],
errors='coerce').dt.year

sns.histplot(patient_df['YearOfBirth'].dropna(), bins=30, kde=True)
plt.title("Patient Year of Birth Distribution")
plt.xlabel("Year")
plt.ylabel("Count")
plt.show()
```

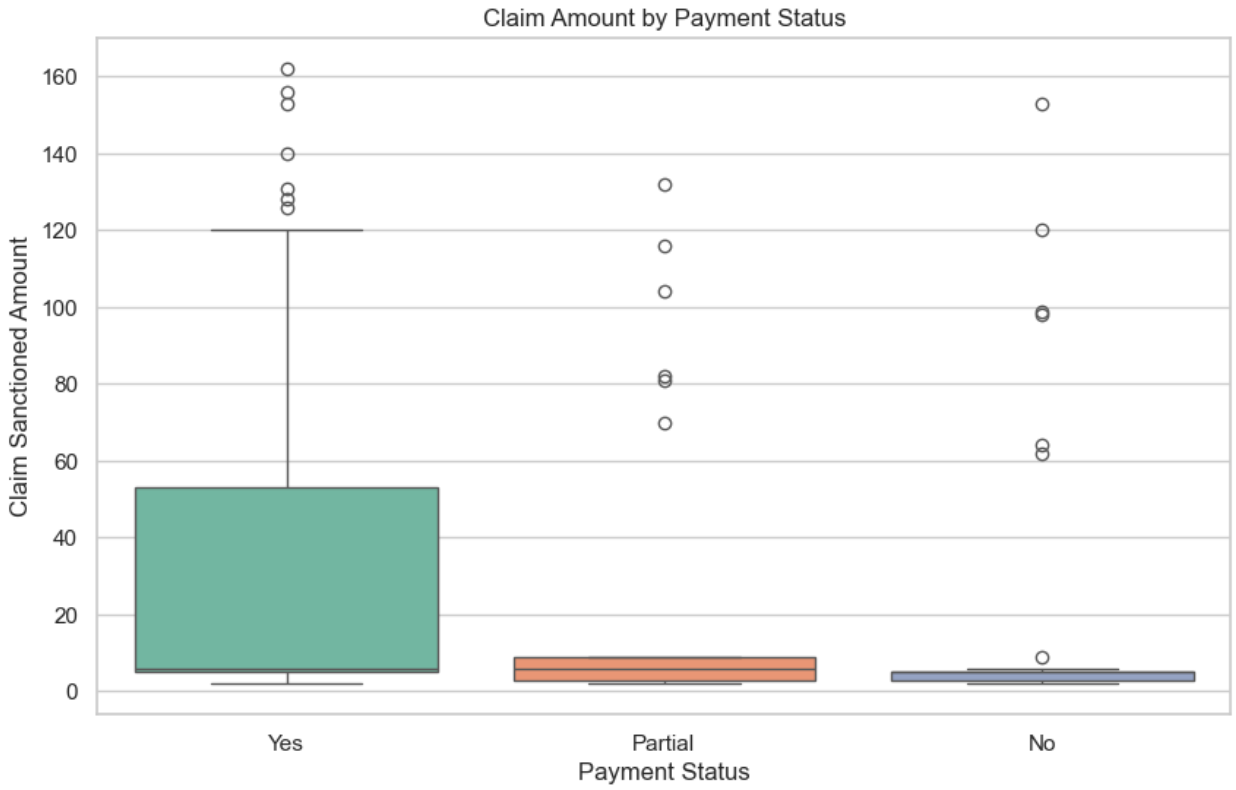



```
sns.boxplot(x='PaymentStatus', y='ClaimSanctionAmt', data=billing_df,
            palette="Set2")  # Try: "viridis", "plasma", "Set1",
"tab10"
plt.title("Claim Amount by Payment Status")
plt.xlabel("Payment Status")
plt.ylabel("Claim Sanctioned Amount")
plt.show()
```

```
C:\Users\MOHIT\AppData\Local\Temp\ipykernel_25404\3545574933.py:1:
FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be
removed in v0.14.0. Assign the `x` variable to `hue` and set
`legend=False` for the same effect.

  sns.boxplot(x='PaymentStatus', y='ClaimSanctionAmt',
data=billing_df,
```



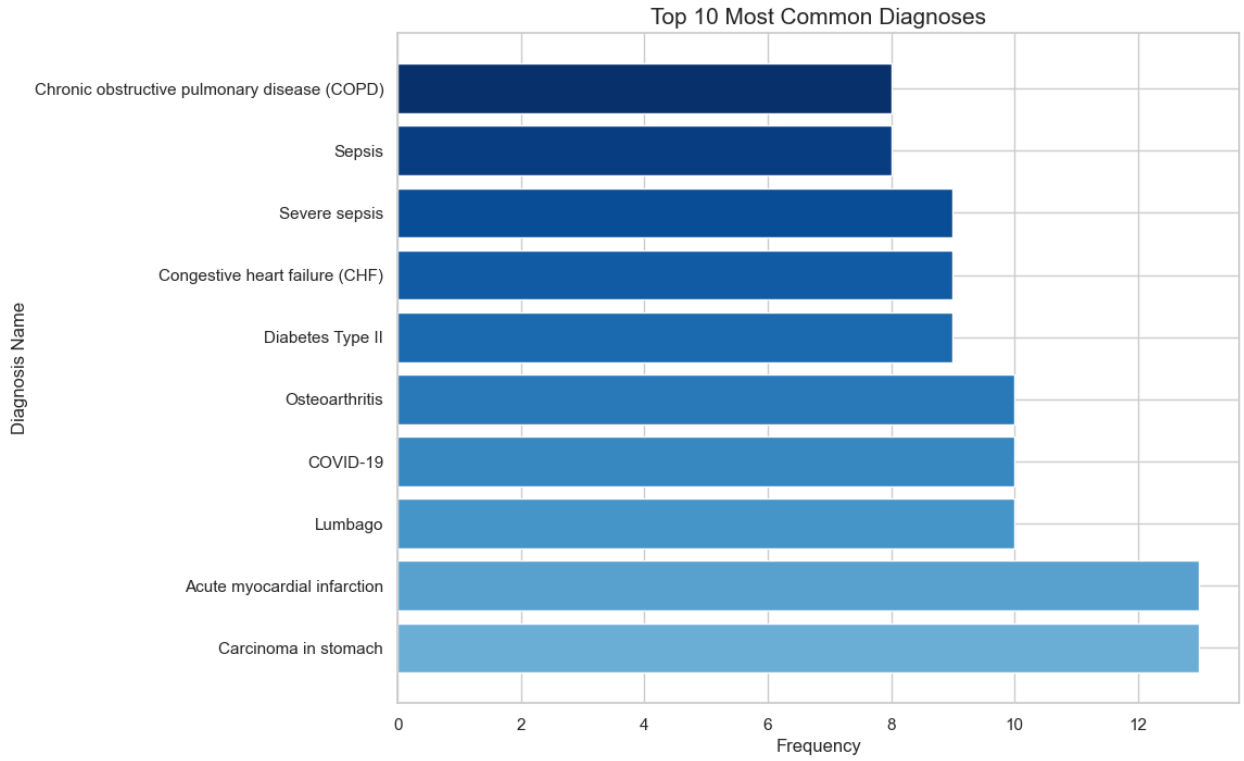

```
print("Average Claim by Payment Status:\n",
billing_df.groupby('PaymentStatus')['ClaimSanctionAmt'].mean())

Average Claim by Payment Status:
 PaymentStatus
No         21.875000
Partial    27.040000
Yes        35.679245
Name: ClaimSanctionAmt, dtype: float64

dx_df = pd.read_sql("""
    SELECT d.DxCode, dd.DxName
    FROM diagnosis d
    JOIN diagnosisdetails dd ON d.DxCode = dd.DxCode
""", engine)
```

```
top_diagnoses = dx_df['DxName'].value_counts().head(10)

# Create a custom color palette that transitions based on frequency
plt.figure(figsize=(10, 8))
bars = plt.barh(top_diagnoses.index, top_diagnoses.values,
                color=plt.cm.Blues(np.linspace(0.5, 1,
len(top_diagnoses))))
plt.title("Top 10 Most Common Diagnoses", fontsize=15)
plt.xlabel("Frequency", fontsize=12)
plt.ylabel("Diagnosis Name", fontsize=12)
plt.show()
```




```
engine.dispose()
```