Hospital Management Data Analysis Project

By: Roshan Nikam

---

1. Introduction

This project focuses on analysing hospital operations using data from various departments including doctors, patients, appointments, billing, and treatments. The goal was to process, query, and visualize the data to uncover actionable insights to support healthcare decisions and improve hospital efficiency.

---

2. Project Objectives

- Import and manage hospital data using PostgreSQL (pgAdmin).
- Execute complex SQL queries for data analysis.
- Visualize the queries and KPIs using Power BI and Python.
- Build an interactive dashboard for quick decision-making.
- Identify patterns, inefficiencies, and revenue metrics.

---

3. Tools & Technologies Used

Tool	Purpose
pgAdmin	Data storage, SQL queries
Power BI	Dashboards, data visualization
Python	Data cleaning, advanced charts
Pandas	SQL processing in Python
Seaborn/Matplotlib	Data plots

---

4. Dataset Overview

worked with 5 datasets in CSV format:

File Name

doctors.csv

patients.csv

appointments.csv

billing.csv

treatment.csv

---

5. Data Preparation and Import

- CSV files were imported into PostgreSQL using pgAdmin.
- Tables were cleaned and relationships were created using foreign keys (e.g., patient ID in billing linked to patients).
- Data types were cast properly (dates, numeric fields, etc.).
- Null values were handled during import.
- Added new columns and created measures in powerbi

---

6. SQL Querying using pgAdmin

Performed SQL analysis using PostgreSQL. Key queries included:

- Total appointments were made per doctor
- patients have the highest number of appointments
- trend of appointments over the last 12 months
- How many appointments were completed, cancelled, or pending
- total billing amount generated per month
- doctor generates the most revenue
- average billing amount per treatment type
- percentage of billing is overdue or unpaid
- patients received each type of treatment
- average cost associated with each treatment
- revenue by each treatment

- doctors have the highest number of appointments
- average appointment time per doctor
- revenue generated by each doctor
- do patients revisit the hospital
- percentage of patients have multiple treatments
- seasonal patterns in patient visits or treatments

Example Query:

```
SELECT d.name AS doctor_name, COUNT(a.appointment_id) AS total_appointments
FROM appointments a
JOIN doctors d ON a.doctor_id = d.doctor_id
GROUP BY d.name;
```

---

7. Power BI Dashboard

- Imported SQL data into Power BI.
- Created dynamic and interactive dashboards with slicers for:
    - Reason for visit
    - Treatment type

Power BI Visuals:

- Bar Chart: doctors attended patients
- Line Chart: Monthly appointments
- Pie Chart: Payment Status
- Donut chart: Appointment status
- Table: Patient revisit
- Total revenue
- Total patients
- Gender distribution

---

8. Python Visualizations

Used Python to create additional insights and visualizations using Pandas, Matplotlib, and Seaborn.

Visuals Created:

- Bar plot's
- Pie chart
- Tables
- line graph
- Example Python Code:

```
import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

query1=""" SELECT d.name AS doctor_name, COUNT(a.appointment_id) AS
total_appointments

FROM appointments a

JOIN doctors d ON a.doctor_id = d.doctor_id

GROUP BY d.name;"""

df = pd.read_sql("query1, engine")

sns.barplot(data=df, x='department', y='total_charges')

plt.title("Billing Charges by Department")

plt.show()
```

---

9. Key Insights & Results

- Highest patient traffic is in the **Chemotherapy** and **X-Ray treatment.**
- **25% Appointment Cancelled**
- Doctor **Dr. Sarah Taylor** had the highest number of patients.
- **Quarter two of the year and first month** had peak appointment bookings.

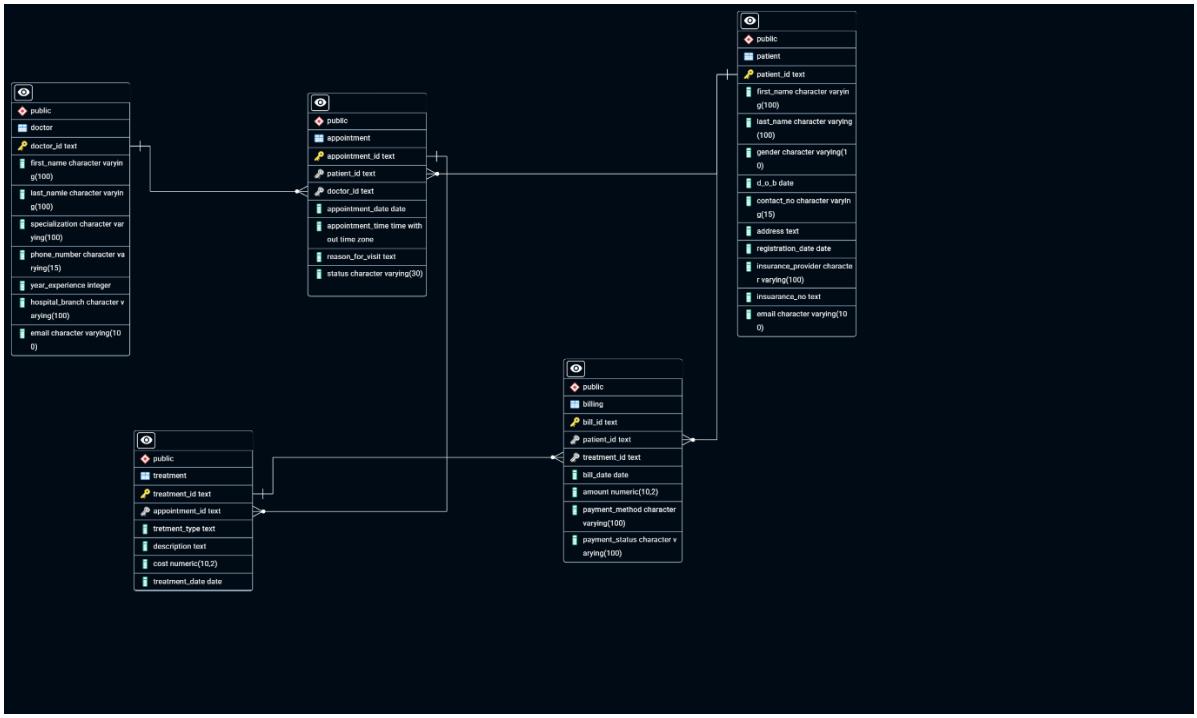
- **34%** billing amount pending and **33%** bill failed.
- Generated most revenue in **first month.**
- **16-time** one patient took appointments.
- Generated most revenue in **MRI treatment type.**
- **38 %** received payment by **credit card**
- **10-time** one patient visited to hospital.

11. Conclusion

The project effectively demonstrated how hospital data can be transformed into insights through SQL, Power BI, and Python. The dashboards and visualizations provide a useful analytical tool for hospital administrators to monitor operations and improve service delivery.

14. Appendix

- **ERD**



- **SQL Query Snapshots**

```sql
--- create table Doctor
create table Doctor(
                    Doctor_id TEXT primary key ,
                    First_Name VARCHAR(100) not null,
                    Last_Namle VARCHAR(100),
                    Specialization VARCHAR(100),
                    Phone_Number VARCHAR(15) ,
                    Year_Experience INT,
                    Hospital_Branch VARCHAR(100),
                    Email VARCHAR(100)

);
---create table Patient
create table Patient(
                    Patient_id TEXT primary key,
                    First_Name VARCHAR(100) not null,
                    Last_name VARCHAR(100),
                    Gender VARCHAR(10),
                    D_o_b DATE,
                    Contact_No VARCHAR(15),
                    Address TEXT,
                    Registration_date DATE,
                    Insurance_Provider VARCHAR(100),
                    Insuarance_No TEXT,
```

```sql
--- create table Appointment
create table Appointment(
                        Appointment_id TEXT primary key,
                        Patient_id TEXT REFERENCES Patient(Patient_id),
                        Doctor_id TEXT REFERENCES Doctor(Doctor_id),
                        Appointment_date DATE,
                        Appointment_time TIME,
                        ReAson_for_visit TEXT,
                        Status VARCHAR(30)
);
---create tabke tretment
create table Treatment(
                        Treatment_id TEXT primary key,
                        Appointment_id TEXT REFERENCES Appointment(Appointment_id),
                        Tretment_type TEXT,
                        Description TEXT,
                        Cost DECIMAL(10,2),
                        Treatment_date DATE

);
---create table billing
create table Billing(
                        Bill_id TEXT primary key,
                        Patient_id TEXT REFERENCES Patient(Patient_id),
```

```sql
---- add data from csv
copy doctor from 'D:\postgres\Projects\hospital_management\doctors.csv' delimiter ',' HEADER;
copy patient from 'D:\postgres\Projects\hospital_management\patients.csv' delimiter ',' HEADER;
copy appointment from 'D:\postgres\Projects\hospital_management\appointments.csv' delimiter ',' HEADER;
copy treatment from 'D:\postgres\Projects\hospital_management\treatments.csv' delimiter ',' HEADER;
copy billing from 'D:\postgres\Projects\hospital_management\billing.csv' delimiter ',' HEADER;

---
```

```sql
---
---Patient & Appointment Analysis

---1.How many total appointments were made per doctor?
select d.first_name ||' ' || d.last_namle as doctor_name,
count(a.appointment_id) as total_appointment from doctor as d
left join appointment as a  on d.doctor_id=a.doctor_id
group by doctor_name
order by total_appointment desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 10 | Page No: 1 of 1

	doctor_name text	total_appointment bigint
1	Sarah Taylor	29
2	David Taylor	25
3	Alex Davis	24

```sql
---2.Which patients have the highest number of appointments?
select p.first_name ||' '|| p.last_name as patient_name,
count(a.appointment_id) as total_appointments from patient as p
left join appointment as a on p.patient_id=a.patient_id
group by patient_name
order by total_appointments desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 39 | Page No: 1 of 1

	patient_name text	total_appointments bigint
1	Michael Taylor	16
2	David Wilson	15
3	Laura Davis	14

```sql
---3.What is the trend of appointments over the last 12 months?
SELECT  extract(month from appointment_date) AS month_name,
          COUNT(*) AS total_appointments
FROM appointment group by month_name order by month_name;
```

Data Output | Messages | Notifications

Showing rows: 1 to 12 | Page No: 1 of 1

	month_name numeric	total_appointments bigint
1	1	20
2	2	14
3	3	19

```sql
---4.How many appointments were completed, canceled, or pending?
select status,count(a.appointment_id) as appointment from appointment as a
group by status order by appointment desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	status character varying (30)	appointment bigint
1	No-show	52
2	Scheduled	51
3	Cancelled	51

```sql
---Billing & Revenue

---5.What is the total billing amount generated per month?
select extract(month from b.bill_date)as Month ,sum(b.amount) as Total_Amount
from billing as b group by Month order by Month ;
```

Data Output | Messages | Notifications

Showing rows: 1 to 12 | Page No: 1 of 1

	month numeric	total_amount numeric
1	1	58701.23
2	2	36669.69
3	3	47304.29

```sql
--- 6.Which doctor generates the most revenue?
select d.first_name ||' '|| d.last_namle as doctor_name ,
sum(b.amount) as revenue from doctor as d
join appointment as a on d.doctor_id=a.doctor_id
join billing as b on a.patient_id=b.patient_id
group by doctor_name order by revenue desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 10 | Page No: 1 of 1

	doctor_name text	revenue numeric
1	Sarah Taylor	394356.89
2	Alex Davis	339280.84
3	David Taylor	336851.92

```sql
---7.What is the average billing amount per treatment type?
select  t.tretment_type as treatmenttype ,avg(b.amount) as average_amount
from treatment as t
left join billing as b on t.treatment_id=b.treatment_id
group by treatmenttype order by average_amount desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 5 | Page No: 1 of 1

	treatmenttype text	average_amount numeric
1	MRI	3224.9488888888888889
2	Physiotherapy	2761.6138888888888889
3	X-Ray	2698.8700000000000000

```sql
---8.What percentage of billing is overdue or unpaid?
select payment_status,count(*)*100.0/(select count(*) from billing) as percentage
   from billing group by Payment_status;
```

Data Output | Messages | Notifications

Showing rows: 1 to 3 | Page No: 1 of 1

	payment_status character varying (100)	percentage numeric
1	Paid	32.0000000000000000
2	Failed	33.5000000000000000
3	Pending	34.5000000000000000

```sql
---9.How many patients received each type of treatment?
select tretment_type,count(appointment_id) as total
from treatment group by tretment_type order by total desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 5 | Page No: 1 of 1

	tretment_type text	total bigint
1	Chemotherapy	49
2	X-Ray	41
3	ECG	38

```sql
---10.What is the average cost associated with each treatment?
select tretment_type, avg(cost) as average_cost from treatment
group by tretment_type order by average_cost desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 5 | Page No: 1 of 1

	tretment_type text	average_cost numeric
1	MRI	3224.9488888888888889
2	Physiotherapy	2761.6138888888888889
3	X-Ray	2698.8700000000000000

```sql
---11 what is the revenue by each treatment?
select tretment_type, sum(cost) as revenue from treatment group by tretment_type
order by revenue desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 5 | Page No: 1 of 1

	tretment_type text	revenue numeric
1	Chemotherapy	128855.68
2	MRI	116098.16
3	X-Ray	110653.67

```sql
---Doctor Performance

---12.Which doctors have the highest number of appointments?
select d.first_name ||' '||d.last_namle  as Name ,count(a.appointment_id) as total from doctor as d
left join appointment as a on d.doctor_id=a.doctor_id
group by Name order by total desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 10 | Page No: 1 of 1

	name text	total bigint
1	Sarah Taylor	29
2	David Taylor	25
3	Alex Davis	24

```sql
---13.What is the average appointment time per doctor? (if you have time data)
select d.first_name ||' '|| d.last_namle as name , avg(a.appointment_time) as average_time
from doctor as d
left join appointment as a on d.doctor_id=a.doctor_id
group by name order by average_time desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 10 | Page No: 1 of 1

	name text	average_time interval
1	Jane Smith	14:05:27.272727
2	Linda Wils...	13:30:00
3	Sarah Smith	13:13:14.117647

```sql
---14.How does the revenue generated by each doctor compare?
select d.first_name ||' '|| d.last_namle as name ,sum(b.amount) as total_revenue from doctor as d
join appointment as a on d.doctor_id=a.doctor_id
join billing as b on a.patient_id=b.patient_id
group by name order by total_revenue desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 10 | Page No: 1 of 1

	name text	total_revenue numeric
1	Sarah Taylor	394356.89
2	Alex Davis	339280.84
3	David Taylor	336851.92

```sql
---Patient Behavior & Outcomes

---15.How often do patients revisit the hospital ?
select distinct(p.patient_id) as patientID ,
p.first_name ||' '|| p.last_name as name ,
count(a.appointment_id) as visit_total from patient as p
join appointment as a on p.patient_id=a.patient_id
group by patientID , name order by visit_total desc;
```

Data Output | Messages | Notifications

Showing rows: 1 to 48 | Page No: 1 of 1

	patientid text	name text	visit_total bigint
1	P012	Laura Davis	10
2	P005	David Wilson	8
3	P016	Michael Taylor	7

```sql
---16.What percentage of patients have multiple treatments?
WITH treatment_counts AS (
    SELECT
        a.patient_id,
        COUNT(t.treatment_id) AS treatment_count
    FROM treatment t
    JOIN appointment a ON t.appointment_id = a.appointment_id
    GROUP BY a.patient_id
)
SELECT
    CASE WHEN treatment_count > 1 THEN 'Multiple Treatments' ELSE 'Single Treatment' END AS treatment_group,
    COUNT(*) * 100.0 / (SELECT COUNT(DISTINCT patient_id) FROM appointment) AS percentage
FROM treatment_counts
GROUP BY treatment_group;
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

	treatment_group text	percentage numeric
1	Multiple Treatments	100.0000000000000000

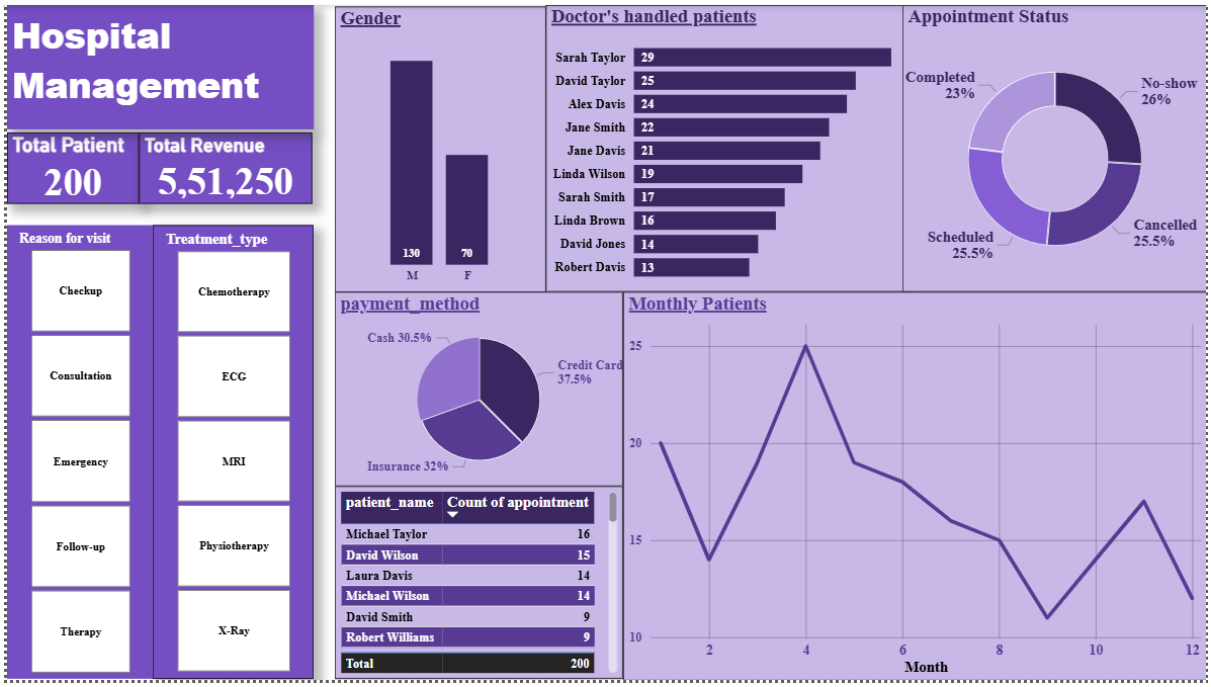
```sql
---17.Are there seasonal patterns in patient visits or treatments?
with seasonal as( select extract(month from treatment_date)as month ,
count(*) as total_appointment from treatment group by month order by month)
select
      case
      when month>=1 and month<=3 then 'quarter_1'
      when month>=3 and month<=6 then 'quarter_2'
      when month>=6 and month<=9 then 'quarter_3'
      when month>=9 and month <=12 then 'quarter_4'
      else 'invalid' end as quarters,
      sum(total_appointment) as total_appointment from  seasonal
      group by quarters order by quarters ;

---- another way
```

Data Output | Messages | Notifications

Showing rows: 1 to 4 | Page No: 1 of 1

	quarters text	total_appointment numeric
1	quarter_1	53
2	quarter_2	62
3	quarter_3	42

- **Power BI Dashboard Images**



- **Python Graph Outputs**

1)

```
tables = pd.read_sql("SELECT table_name FROM information_schema.tables WHERE table_schema='public';", engine)
print("Tables in database:")
print(tables)
```

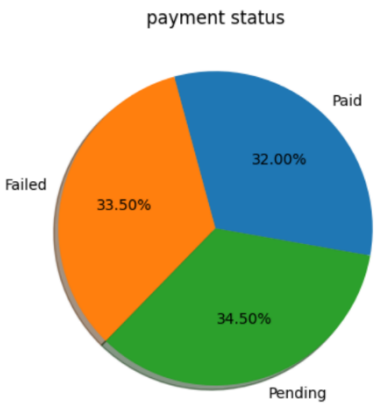
```
Tables in database:
    table_name
0      patient
1  appointment
2       doctor
3    treatment
4      billing
```

```
[2]: query=""" select * from Doctor """
df1=pd.read_sql(query,engine)
df1.head()
```

[2]:

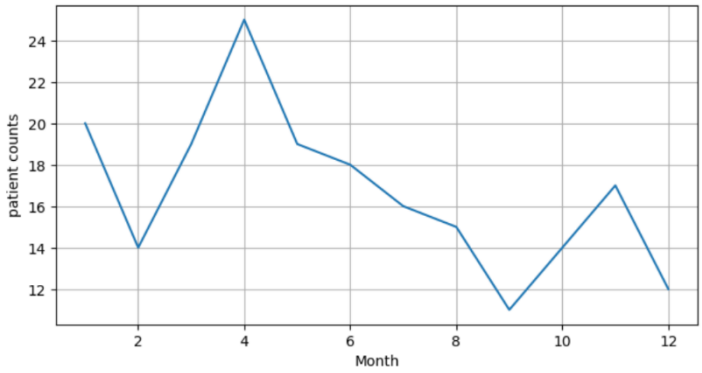
	doctor_id	first_name	last_name	specialization	phone_number	year_experience	hospital_branch	email
0	D001	David	Taylor	Dermatology	8322010158	17	Westside Clinic	dr.david.taylor@hospital.com
1	D002	Jane	Davis	Pediatrics	9004382050	24	Eastside Clinic	dr.jane.davis@hospital.com
2	D003	Jane	Smith	Pediatrics	8737740598	19	Eastside Clinic	dr.jane.smith@hospital.com
3	D004	David	Jones	Pediatrics	6594221991	28	Central Hospital	dr.david.jones@hospital.com
4	D005	Sarah	Taylor	Dermatology	9118538547	26	Central Hospital	dr.sarah.taylor@hospital.com

2)

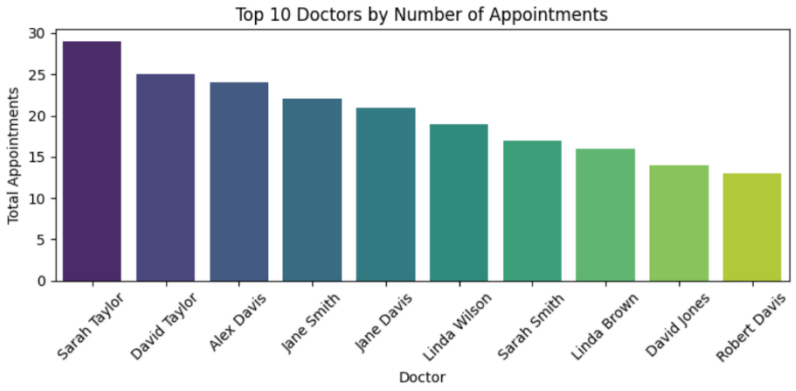


3)

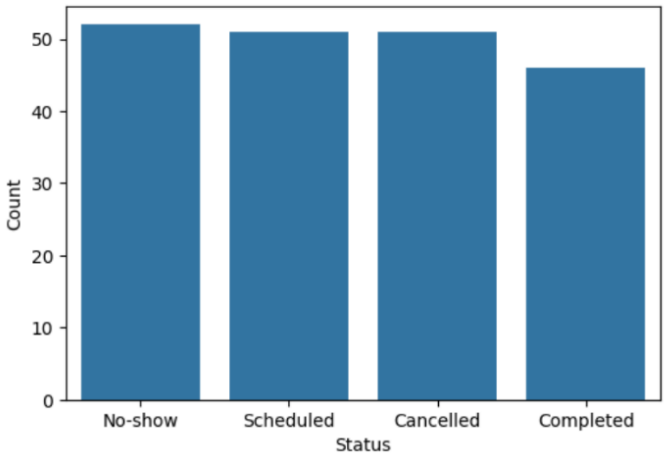
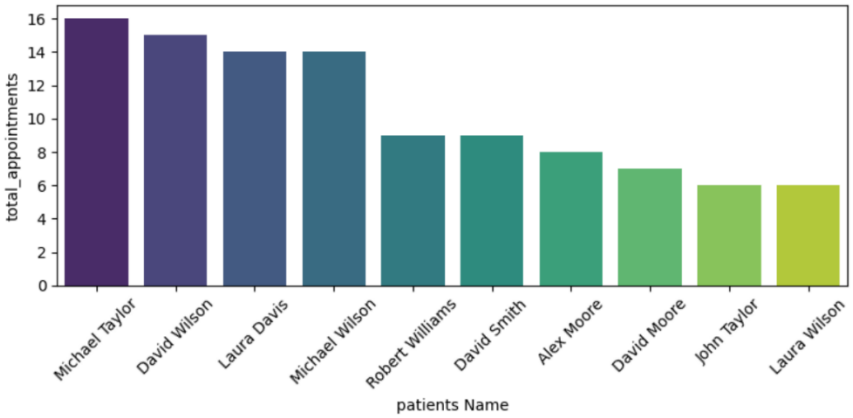
```
C:\Users\ADMIN\AppData\Local\Temp\ipykernel_4704\1711229506.py:9: UserWarning: Ignoring `palette` bec
  sns.lineplot(data=df_month , x='month_name',y='total_appointments' ,palette='viridis')
```



4)



5)



6)