



Faucet Medical Hospital

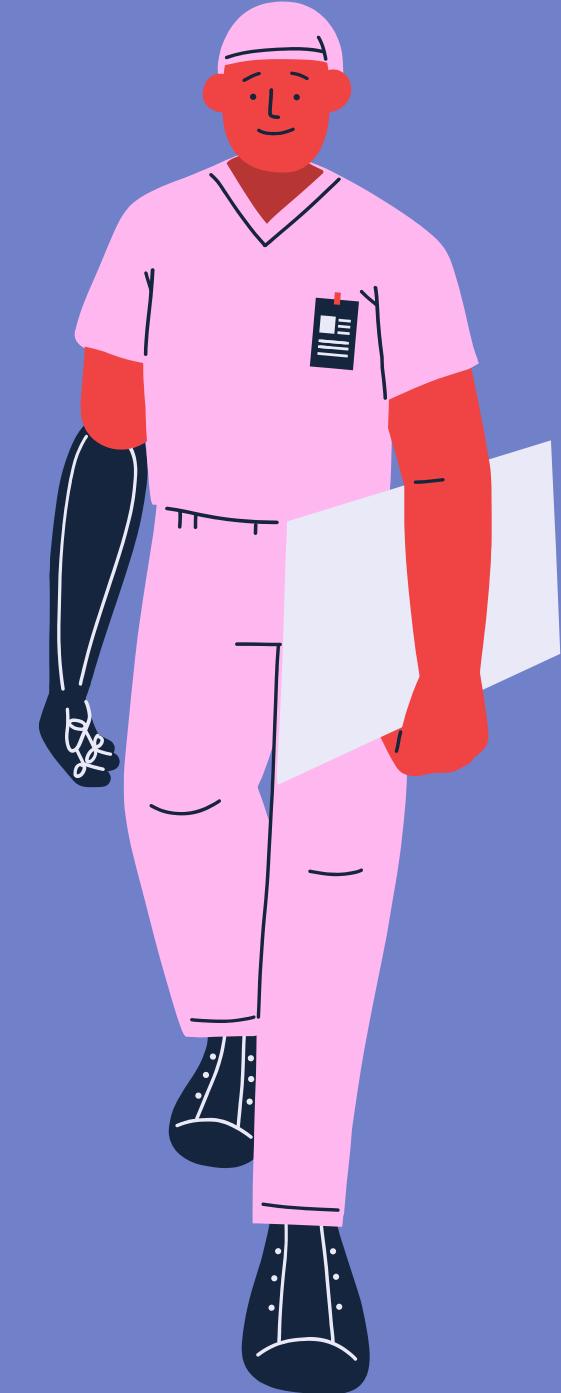
# Hospital Management Analysis with SQL

Presented By - Gitanjali Pekamwar



# Project Overview

The Hospital Management Analysis project aims to analyze and optimize healthcare operations using SQL. This project provides valuable insights into patient appointments, doctor performance, prescription trends, and hospital efficiency and enhance decision-making processes, by querying a structured hospital database.



# Table Creation:

```
-- Doctors Table
CREATE TABLE doctors (
    doctor_id INT PRIMARY KEY,
    first_name VARCHAR(50),
    last_name VARCHAR(50),
    specialty VARCHAR(50),
    contact_number VARCHAR(15)

-- Prescriptions Table
CREATE TABLE prescriptions (
    prescription_id INT PRIMARY KEY,
    appointment_id INT,
    medication VARCHAR(50),
    dosage VARCHAR(20),
    instructions VARCHAR(50),
    FOREIGN KEY (appointment_id)
    REFERENCES appointments(appointment_id)
);
```

```
1 •  create database Hospital_Management_db;
2 •  use Hospital_Management_db;
3
4   -- Patients Table
5 •  CREATE TABLE patients (
6       patient_id INT PRIMARY KEY,
7       first_name VARCHAR(50),
8       last_name VARCHAR(50),
9       gender VARCHAR(10),
10      age INT,
11      contact_number VARCHAR(15)
12  );

-- Appointments Table
CREATE TABLE appointments (
    appointment_id INT PRIMARY KEY,
    patient_id INT,
    doctor_id INT,
    appointment_date DATE,
    status VARCHAR(20),
    FOREIGN KEY (patient_id) REFERENCES patients(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES doctors(doctor_id)
```

# I)Find the average age of patients for each gender.

```
-- 1)Find the average age of patients for each gender.  
select gender,  
       avg(age) as Patients_avg_age  
  from patients  
group by gender;
```

	gender	Patients_avg_age
▶	Male	48.9283
	Female	49.4360



# 2) List the top 5 doctors with the highest number of completed appointments.

```
-- 2)List the top 5 doctors with the highest number of completed appointments.  
select  
    d.first_name as Doctors_FirstName,  
    d.last_name as Doctors_LastName,  
    DATE_FORMAT(appointment_date, '%Y-%m') as month,  
    COUNT(*) as total_appointments  
from  
    appointments a  
join doctors d  
        on a.doctor_id = d.doctor_id  
group by Doctors_FirstName,  
    Doctors_LastName,  
    DATE_FORMAT(appointment_date, '%Y-%m')  
order by  
    month;
```

	Doctors_FirstName	Doctors_LastName	month	total_appointments
▶	Dr. Michael	Doe	2024-01	3
	Dr. Sarah	Johnson	2024-01	3
	Dr. Michael	Williams	2024-01	1
	Dr. David	Williams	2024-01	2
	Dr. Emily	Smith	2024-01	3
	Dr. John	Brown	2024-01	1
	Dr. David	Doe	2024-01	1



### 3)Find the most commonly prescribed medication .

```
-- 3)Find the most commonly prescribed medication.  
select  
    medication,  
    COUNT(*) as prescription_count  
from prescriptions  
group by medication  
order by prescription_count DESC  
limit 1;
```

	medication	prescription_count
▶	Metformin	326



# 4) Find doctors having specialty in Dermatology

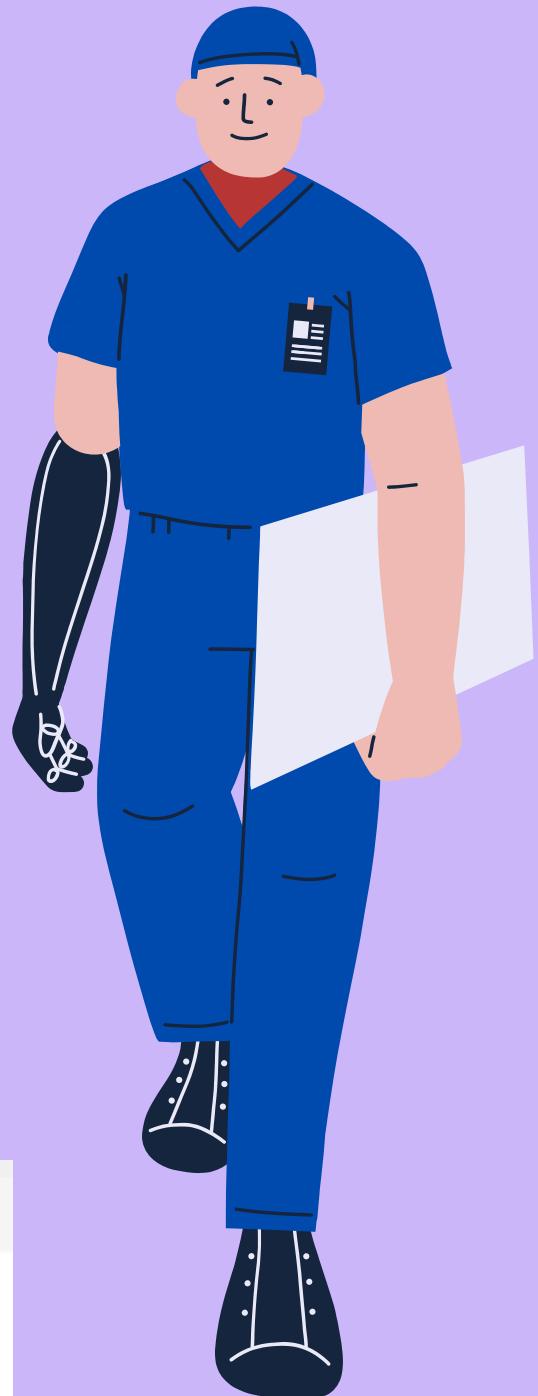
```
-- 4) Find doctors having specialty in Dermatology

select first_name,
       last_name,
       specialty
  from doctors
 where specialty = 'Dermatology';
```

	first_name	last_name	specialty
1	Dr. Emily	Williams	Dermatology
2	Dr. David	Williams	Dermatology
3	Dr. John	Smith	Dermatology
4	Dr. Emily	Johnson	Dermatology
5	Dr. Sarah	Smith	Dermatology
6	Dr. Michael	Doe	Dermatology

# 5)Calculate the percentage of cancelled appointments for each doctor.

```
-- 5)Calculate the percentage of cancelled appointments for each doctor.  
SELECT d.doctor_id,  
       d.first_name,  
       d.last_name,  
       COUNT(case when a.status = 'Cancelled' then  
           1 else NULL end) * 100.0 / COUNT(*) as cancel_percentage  
FROM doctors d  
JOIN  
     appointments a ON d.doctor_id = a.doctor_id  
GROUP BY d.doctor_id, d.first_name, d.last_name  
ORDER BY cancel_percentage DESC;
```



	doctor_id	first_name	last_name	cancel_percentage
▶	3	Dr. Emily	Williams	55.55556
	44	Dr. Michael	Doe	44.44444
	49	Dr. Emily	Johnson	41.30435
	39	Dr. Emily	Johnson	41.30435
	25	Dr. Sarah	Johnson	40.47619

# 6)Find the top 5 patients who have made the most appointments.

```
-- 6)Find the top 5 patients who have made the most appointments.  
select p.first_name as Patients_FirstName,  
       p.last_name as Patients_LaststName,  
       count(a.appointment_id) as Total_appointments  
from patients p  
join appointments a on p.patient_id = a.patient_id  
group by Patients_FirstName,  
         Patients_LaststName  
order by Total_appointments desc  
Limit 5 ;
```

	Patients_FirstName	Patients_LaststName	Total_appointments
	David	Anderson	43
	Bob	Taylor	40
	Sophia	Garcia	39
	Alice	Brown	39
	Bob	Williams	37

# 7)Calculate the average number of appointments completed for a doctor.

```
-- 7)Calculate the average number of appointments completed for a doctor.  
select  
    avg(Completed_appointments) as Avg_completed_appointments_per_doctor  
  from  
(select  
    doctor_id,  
    count(*) as Completed_appointments  
  from appointments  
    where status = 'Completed'  
group by doctor_id)  
    as Completed_data;
```

	Avg_completed_appointments
▶	13.4400



# 8)List the medications that were prescribed more than 5 times in a month.

```
-- 8)List the medications that were prescribed more than 5 times in a month.  
select p.medication as Medication,  
       date_format(a.appointment_date, '%y-%m') as Month,  
       count(*) as Priscription_Count  
from prescriptions p  
      Join appointments a on p.appointment_id = a.appointment_id  
      group by p.medication, Month  
having count(*) > 5  
      order by Priscription_Count desc;
```

	Medication	Month	Priscription_Count
▶	Ibuprofen	24-08	10
	Amoxicillin	25-11	10
	Ibuprofen	27-10	10
	Atorvastatin	26-11	9
	Metformin	26-09	9
	Metformin	26-07	9
	Metformin	24-04	9

# 9)Find the percentage of completed, canceled, and scheduled appointments for each month.

```
-- 9)Find the percentage of completed, canceled, and scheduled appointments  
-- for each month.  
  
select date_format(appointment_date, '%Y-%M') as Month,  
sum(case when status = 'completed' then 1 else 0 end)*100/count(*) as completed_percent  
sum(case when status = 'cancelled' then 1 else 0 end)*100/count(*) as cancelled_percent  
sum(case when status = 'Scheduled' then 1 else 0 end)*100/count(*) as Scheduled_percent  
from appointments  
group by Month ;
```

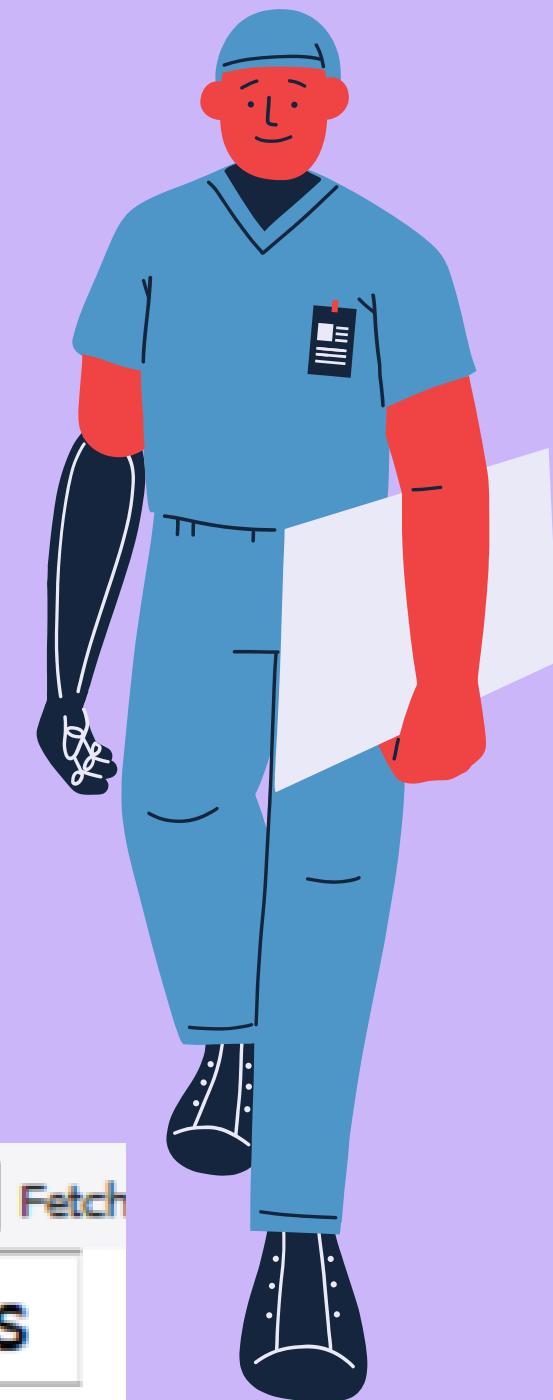
Month	completed_percentage	cancelled_percentage	Scheduled_percentage
2024-January	32.2581	38.7097	29.0323
2024-February	34.4828	27.5862	37.9310
2024-March	22.5806	54.8387	22.5806
2024-April	36.6667	26.6667	36.6667
2024-May	35.4839	38.7097	25.8065
2024-June	20.0000	33.3333	46.6667



# 10) Identify which doctor has attended to the highest number of unique patients.

```
-- 10) Identify which doctor has attended to the highest number of unique patients.  
select distinct d.first_name as Doctors_FirstName,  
           d.last_name as Doctors_LastName,  
       count(*) as Unique_Patients  
  from appointments a  
join patients p on a.patient_id = p.patient_id  
join doctors d on d.doctor_id = a.doctor_id  
group by d.first_name,d.last_name  
order by Unique_Patients desc  
limit 1;
```

	Doctors_FirstName	Doctors_LastName	Unique_Patients
▶	Dr. Emily	Johnson	208



# 11)Find the number of patients who have visited the hospital more than once in the past 6 months.

```
-- 11)Find the number of patients who have visited  
-- the hospital more than once in the past 6 months.  
  
select  
    p.first_name as Patients_FirstName,  
    p.last_name as Patients_LastName,  
    COUNT(*) as visit_count  
from appointments a  
join patients p on a.patient_id = p.patient_id  
where  
    appointment_date >= DATE_SUB(CURDATE(), INTERVAL 6 MONTH)  
group by Patients_FirstName,Patients_LastName  
having COUNT(*) > 1;
```

	Patients_FirstName	Patients_LastName	visit_count
▶	John	Smith	24
	Sarah	Miller	13
	Alice	Brown	35
	Bob	Johnson	25
	Laura	Taylor	11
	Bob	Taylor	35
	Laura	Garcia	10

# 12) Find the Most Recent Appointment for among the Patients Using Subquery.

```
-- 12) Find the Most Recent Appointment for among the Patients Using Subquery.  
select *  
  from appointments a  
 where appointment_date =  
 (select  
      max(appointment_date)  
    from appointments a  
   where patient_id = a.patient_id  
 );
```

	appointment_id	patient_id	doctor_id	appointment_date	status
▶	2000	404	15	2029-06-22	Completed



# 13) Calculate the Running Total of Appointments Per Month Using Window Functions.

```
-- 13)Calculate the Running Total of Appointments Per Month Using Window Functions.  
select  
    DATE_FORMAT(appointment_date, '%Y-%m') as month,  
    COUNT(*) as total_appointments,  
    SUM(COUNT(*)) over (ORDER BY DATE_FORMAT(appointment_date, '%Y-%m')) as running_total  
from  
    appointments  
group by month  
order by running_total DESC;
```

	month	total_appointments	running_total
▶	2029-06	22	2000
	2029-05	31	1978
	2029-04	30	1947
	2029-03	31	1917
	2029-02	28	1886
	2029-01	31	1858

# 14)Find the First and Second Most Prescribed Medications Using Dense Rank

```
-- 14)Find the First and Second Most Prescribed Medications Using Dense Rank

select
    medication,
    COUNT(*) as prescription_count,
    dense_rank() over (ORDER BY COUNT(*) DESC) as Rank_number
from
    prescriptions
group by
    medication
LIMIT 2;
```

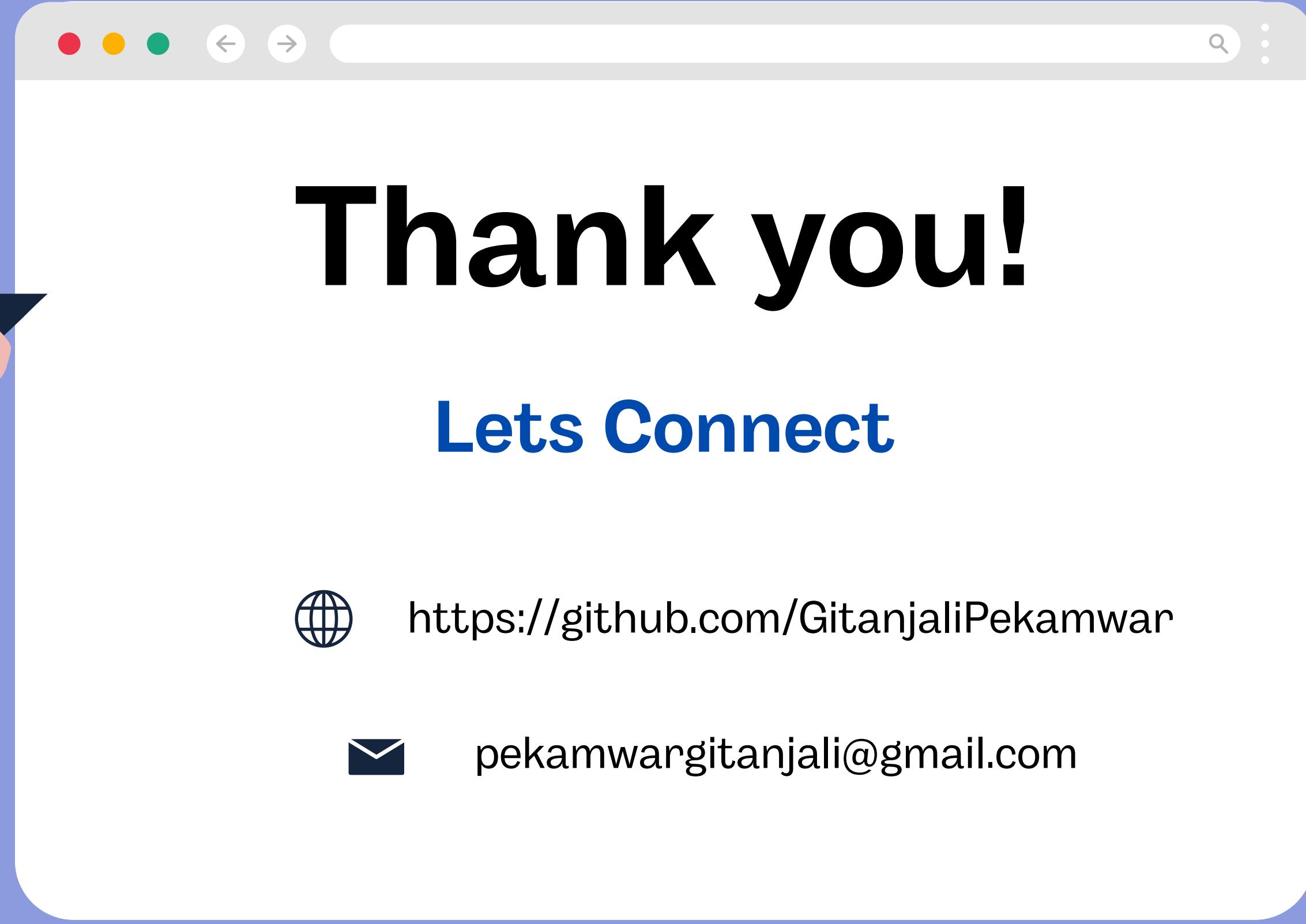
	medication	prescription_count	Rank_number
▶	Metformin	326	1
	Atorvastatin	302	2

# Insights

- Appointment Trends Analysis: Helped identify peak consultation times and patient appointment behavior
- Doctor Performance Ranking: Assisted in workload distribution and performance optimization.
- Prescription Insights: Revealed the most commonly used medications, aiding stock management.
- Hospital Efficiency: Showcased retention rates and patient revisit patterns for better operational planning.

# Conclusion

The Hospital Management Analysis using SQL project helped me understand hospital operations by analyzing patient appointments, doctor performance, prescriptions, and patients behavior. I used basic to advanced SQL concepts like SELECT, WHERE, ORDER BY, and GROUP BY to retrieve and summarize data. Joins were used to combine patient, doctor, and appointment details, while aggregate functions like COUNT, SUM, and AVG helped in data analysis. I applied subqueries to fetch recent patient visits and window functions (RANK, DENSE\_RANK) to rank doctors and track trends. Time-based functions (DATE\_FORMAT, CURDATE) helped in analyzing appointment patterns. This project showed how SQL can be used to extract valuable insights for better hospital management, and it can be expanded further with data visualization and predictive analytics for even deeper insights.



<https://github.com/GitanjaliPekamwar>



pekamwargitanjali@gmail.com