Problem Statement -

The healthcare industry often deals with complex data involving patient demographics, admission trends, and doctor performance, which are crucial for making informed business and clinical decisions. To assist our hospital in enhancing patient care, operational efficiency, and strategic planning, we conducted a detailed analysis of patient and admission data. The objective of this project is to uncover insights related to patient demographics, analyze admission patterns, and evaluate doctor performance to support data-driven decision-making.

Project Description -

This project focuses on analyzing a hospital's patient and admission data to extract actionable insights. We used SQL and Python to explore key areas, such as patient demographics, admission trends, and doctor efficiency. The insights derived from these analyses can help hospital management and staff improve resource allocation, understand patient profiles, and streamline hospital operations.

The following queries represent the specific objectives we set out to address in this project. Each query is framed as a business question, aimed at improving the hospital's operational and service standards.

Queries and Analysis -

1. Can you provide the first and last names of all male patients ?

```
cursor.execute(''' SELECT first_name, last_name, gender
                   FROM patients
                   WHERE gender LIKE 'M'; ''')
```

2. Who are the patients that do not have any allergies recorded ?

```
cursor.execute(''' SELECT first_name, last_name FROM patients
                   WHERE allergies IS NULL; ''')
```

3. Could you list the patients whose names start with the letter 'C' ?

```
cursor.execute(''' SELECT * FROM patients
                   WHERE first_name LIKE 'C%';  ''')
```

4. Can we get a list of patients weighing between 100 and 120 units ?

```
cursor.execute(''' SELECT first_name, last_name
                   FROM patients
                   WHERE weight BETWEEN 100 AND 120;  ''')
```

5. Could you update records so that patients without allergy information show as 'No Known Allergies'?

```
cursor.execute(''' UPDATE patients
                   SET allergies = 'NKA'
                   WHERE allergies IS NULL; ''')
```

6. Can we display the full name of each patient ?

```
cursor.execute(''' SELECT CONCAT(first_name, ' ', last_name) AS full_name
                   FROM patients;  ''')
```

7. Could we see each patient's name along with their full province name ?

```
cursor.execute(''' SELECT first_name, last_name, province_name
                   FROM patients AS p
                   INNER JOIN province_names AS pn
                   ON p.province_id = pn.province_id ;  ''')
```

8. How many of our patients were born in 2010 ?

```
cursor.execute(''' SELECT COUNT(birth_date) AS Count
                   FROM patients
                   WHERE strftime('%Y', birth_date) = '2010';  ''')
```

9. Find the tallest patient, showing their full name and height.

```
cursor.execute(''' SELECT first_name, last_name, MAX(height)
                   FROM patients ''')
```

10. Could you retrieve all the details for patients with IDs 1, 45, 534, 879, and 1000 ?

```
cursor.execute(''' SELECT *
                   FROM patients
                   WHERE patient_id IN (1, 45, 534, 879, 1000);  ''')
```

11. What is the total count of admissions to date ?

```
cursor.execute(''' SELECT count(*) AS total_admissions
                   FROM admissions;  ''')
```

12. Who were admitted and discharged on the same day ?

```
cursor.execute(''' SELECT *
                   FROM admissions
                   WHERE admission_date = discharge_date;  ''')
```

13. How many admissions has patient ID 579 had so far ?

```
cursor.execute(''' SELECT patient_id, COUNT(*) AS total_admissions
                   FROM admissions
                   WHERE patient_id = 579;  ''')
```

14. Could you list the unique cities in Nova Scotia where our patients live ?

```
cursor.execute(''' SELECT DISTINCT city AS unique_cities
                   FROM patients
                   WHERE province_id = 'NS';  ''')
```

15. Could you find patients First name, Last name, Birth Date who are over 160 cm in height and over 70 kg in weight ?

```
cursor.execute(''' SELECT first_name, last_name, birth_date
                   FROM patients
                   WHERE height > 160 AND weight > 70;  ''')
```

16. Who in Hamilton has recorded allergies ?

```
cursor.execute(''' SELECT first_name, last_name, allergies
                   FROM patients
                   WHERE city = 'Hamilton' AND allergies != 'NKA';  ''')
```

17. List all unique patient birth years in ascending order.

```
cursor.execute(''' SELECT DISTINCT strftime('%Y', birth_date) AS birth_year
                   FROM patients
                   ORDER BY birth_date ASC;  ''')
```

18. Are there any names that appear only once in our patient records ?

```
cursor.execute(''' SELECT first_name
                   FROM patients
                   GROUP BY first_name
                   HAVING COUNT(first_name) = 1;  ''')
```

19. Could you find patients whose first names start and end with ‘S’ and have at least six letters ?

```
cursor.execute(''' SELECT patient_id, first_name
                   FROM patients
                   WHERE first_name LIKE 'S%S' AND LENGTH(first_name) >= 6;  ''')
```

20. Can we list patients diagnosed with ‘Dementia’ ?

```
cursor.execute(''' SELECT p.patient_id, p.first_name, p.last_name
                   FROM patients AS p
                   LEFT JOIN admissions AS a
                   ON p.patient_id = a.patient_id
                   WHERE diagnosis = 'Dementia';  ''')
```

21. List all patient names ordered by name length, then alphabetically.

```
cursor.execute(''' SELECT first_name
                   FROM patients
                   ORDER BY LENGTH(first_name) ASC, first_name ASC;  ''')
```

22. What is the count of male and female patients ? Display them side by side.

```
cursor.execute(''' SELECT
                   SUM(gender = 'M') AS male_count,
                   SUM(gender = 'F') AS female_count
                   FROM patients;  ''')
```

23. List patients allergic to Penicillin or Morphine, ordered by allergy type and name.

```
cursor.execute(''' SELECT first_name, last_name, allergies
                   FROM patients
                   WHERE allergies IN ('Penicillin', 'Morphine')
                   ORDER BY allergies ASC, first_name ASC, last_name ASC;  ''')
```

24. Are there any patients admitted multiple times for the same diagnosis ?

```
cursor.execute(''' SELECT patient_id, diagnosis
                   FROM admissions
                   GROUP BY patient_id, diagnosis
                   HAVING COUNT(*) > 1;  ''')
```

25. List of cities and the total number of patients in the city. Order from most to least patients and then by city name ascending.

```
cursor.execute(''' SELECT city, COUNT(*) AS total_number_of_patients
                   FROM patients
                   GROUP BY city
                   ORDER BY COUNT(*) DESC, city ASC;  ''')
```

26. Can we show everyone affiliated with the hospital as either ‘Patient’ or ‘Doctor’ ?

```
cursor.execute(''' SELECT first_name, last_name, 'Patient' AS role FROM patients
                   UNION ALL
                  SELECT first_name, last_name, 'Doctor' AS role FROM doctors;  ''')
```

27. List of all allergies by popularity.

```
cursor.execute(''' SELECT allergies, COUNT(*) AS popularity
                   FROM patients
                   GROUP BY allergies
                   HAVING allergies != 'NKA'
                   ORDER BY popularity DESC;  ''')
```

28. Could we get a list of patients born in the 1970s, starting from the earliest birth date ?

```
cursor.execute(''' SELECT first_name, last_name, birth_date
                   FROM patients
                   WHERE strftime('%Y', birth_date) BETWEEN '1970' AND '1979'
                   ORDER BY birth_date ASC;  ''')
```

29. Can we display patient names in ‘LASTNAME, first name’ format, sorted by first name ?

```
cursor.execute(''' SELECT CONCAT( UPPER(last_name), ',' , LOWER(first_name) ) AS patient_names
                   FROM patients
                   ORDER BY LOWER(first_name) DESC;  ''')
```

30. Which provinces have a combined patient height of 7,000 or more ?

```
cursor.execute(''' SELECT province_id, SUM(height) AS total_height
                   FROM patients
                   GROUP BY province_id
                   HAVING total_height >= 7000;  ''')
```

31. Calculate the weight range for patients with the surname 'Maroni'.

```
cursor.execute(''' SELECT MAX(weight) - MIN(weight)
                   FROM patients
                   WHERE last_name = 'Maroni'; ''')
```

32. List of all the days of the month (1-31) and how many admission_dates occurred on that day. Sort by the day with most admissions to least admissions.

```
cursor.execute(''' SELECT strftime('%d', admission_date) AS day, COUNT(*) AS total_admissions
                   FROM admissions
                   GROUP BY day
                   ORDER BY total_admissions DESC; ''')
```

33. Retrieve details of the latest admission for patient ID 542.

```
cursor.execute(''' SELECT *
                   FROM admissions
                   GROUP BY patient_id
                   HAVING MAX(admission_date) AND patient_id = 542; ''')
```

34. List of patient_id, attending_doctor_id, and diagnosis for admissions that match one of the two criteria:
    - patient_id is an odd number and attending_doctor_id is either 1, 5, or 19.
    - attending_doctor_id contains a 2 and the length of patient_id is 3 characters.

```
cursor.execute(''' SELECT patient_id, attending_doctor_id, diagnosis
                   FROM admissions
                   WHERE ( patient_id % 2 != 0 AND attending_doctor_id IN (1, 5, 19) )  OR
                         ( attending_doctor_id LIKE '%2%' AND LENGTH(patient_id) = 3  ) ; ''')
```

35. For each doctor, how many admissions have they attended to date ?

```
cursor.execute(''' SELECT d.first_name, d.last_name, COUNT(*) AS total_admissions
                   FROM doctors AS d
                   LEFT JOIN admissions AS a
                   ON d.doctor_id = a.attending_doctor_id
                   GROUP BY d.doctor_id; ''')
```

36. For each doctor, display their id, full name, and the first and last admission date they attended.

```
cursor.execute(''' SELECT d.doctor_id, CONCAT(d.first_name,' ', last_name) AS full_name, MIN(a.admission_date) AS first_admission_date,
                       MAX(a.admission_date) AS last_admission_date
                   FROM doctors AS d
                   LEFT JOIN admissions AS a
                   ON d.doctor_id = a.attending_doctor_id
                   GROUP BY d.doctor_id, full_name ; ''')
```

37. What is the count of patients from each province ? Show highest to lowest.

```
cursor.execute(''' SELECT pn.province_name, COUNT(*) AS total_patients
                   FROM province_names AS pn
                   LEFT JOIN patients AS p
                   ON pn.province_id = p.province_id
                   GROUP BY pn.province_name
                   ORDER BY total_patients DESC; ''')
```

38. Could you list each admission with the patient's name, diagnosis, and doctor's name ?

```
cursor.execute(''' SELECT CONCAT(p.first_name, ' ', p.last_name) AS patient_full_name, a.diagnosis,
                          CONCAT(d.first_name, ' ', d.last_name) AS doctor_name
                   FROM patients AS p
                   INNER JOIN admissions AS a
                   ON p.patient_id = a.patient_id
                   LEFT JOIN doctors AS d
                   ON a.attending_doctor_id = d.doctor_id; ''')
```

39. Are there any duplicate patients based on first and last names ?

```
cursor.execute(''' SELECT first_name, last_name, COUNT(*) AS duplicate_patients
                   FROM patients
                   GROUP BY first_name, last_name
                   HAVING COUNT(*) > 1; ''')
```

40. List of patient's names with their height (cm to feet), weight (KG to pounds), birth date, and full gender name.

```
cursor.execute(''' SELECT first_name, last_name, birth_date,
                       ROUND(height / 30.48, 1) AS height_in_feet, ROUND(weight * 2.205, 0) AS weight_in_kg,
                       CASE
                           WHEN gender = 'M' THEN 'Male'
                           WHEN gender = 'F' THEN 'Female'
                       END AS role
                   FROM patients; ''')
```

41. Which patients don't have any admission records ?

```
cursor.execute(''' SELECT patient_id, first_name, last_name
                   FROM patients
                   WHERE patient_id NOT IN ( SELECT patient_id FROM admissions ) ; ''')
```

42. Could we categorize patients by weight groups ? Show the count of patients in each group.
    - For example, if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.

```
cursor.execute(''' SELECT
                       FLOOR(weight / 10) * 10 AS weight_group,
                       COUNT(*) AS total_patients
                   FROM patients
                   GROUP BY weight_group
                   ORDER BY weight_group DESC; ''')
```

43. List of patient_id, weight, height, isObese from the patients table.
    - Display isObese as a boolean 0 or 1.
    - Obese is defined as weight(kg)/(height(m)2) >= 30.

```
cursor.execute(''' SELECT patient_id, weight, height,
                   CASE
                       WHEN ( weight / (height/100.0 * height/100.0) ) >= 30 THEN 1
                       ELSE 0
                   END AS isObese
                   FROM patients; ''')
```

44. Could you show patients diagnosed with 'Epilepsy' by Dr. Lisa and include her specialty ?

```
cursor.execute(''' SELECT p.patient_id AS patient_id, p.first_name AS patient_first_name, p.last_name AS patient_last_name,
                       d.speciality AS attending_doctor_speciality
                   FROM patients AS p
                   INNER JOIN admissions AS a
                   ON p.patient_id = a.patient_id
                   INNER JOIN doctors AS d
                   ON a.attending_doctor_id = d.doctor_id
                   WHERE a.diagnosis = 'Epilepsy' AND d.first_name = 'Lisa'; ''')
```

45. All patients who have gone through admissions, can see their medical documents on our site. Those patients are given a temporary password after their first admission. Show the patient_id and temp_password. The password must be the following, in order:
    - patient_id
    - The numerical length of patient's last_name
    - Year of patient's birth_date

```
cursor.execute(''' SELECT p.patient_id, CONCAT( p.patient_id, LENGTH(p.last_name), strftime('%Y', p.birth_date) ) AS temp_password
                   FROM patients AS p
                   RIGHT JOIN admissions AS a
                   ON p.patient_id = a.patient_id
                   GROUP BY p.patient_id; ''')
```

46. Each admission costs $50 for patients without insurance, and $10 for patients with insurance. All patients with an even patient_id have insurance.
Give each patient a 'Yes' if they have insurance, and a 'No' if they do not have insurance. Add up the admission_total cost for each has_insurance group.

```
cursor.execute(''' SELECT
                       CASE
                           WHEN patient_id % 2 = 0 THEN 'YES'
                           ELSE 'NO'
                       END AS has_insurance,
                       SUM(CASE
                               WHEN patient_id % 2 = 0 THEN 10
                               ELSE 50
                           END) AS cost_after_insurance
                   FROM admissions
                   GROUP BY has_insurance; ''')
```

47. Which provinces have more male than female patients ? Show the province names.

```
cursor.execute(''' SELECT pn.province_name
                   FROM patients AS p
                   INNER JOIN province_names AS pn
                   ON p.province_id = pn.province_id
                   GROUP BY province_name
                   HAVING
                       COUNT( CASE WHEN gender = 'M' THEN 1 END ) > COUNT( CASE WHEN gender = 'F' THEN 1 END ); ''')
```

48. We are looking for a specific patient. Pull all columns for the patient who matches the following criteria:
    - first_name contains an 'r' after the first two letters.
    - Identifies their gender as 'F'.
    - Born in February, May, or December.
    - Their weight would be between 60kg and 80kg.
    - Their patient_id is an odd number.
    - They are from the city 'Kingston'.

```
cursor.execute(''' SELECT *
                   FROM patients
                   WHERE first_name LIKE '__r%'
                       AND gender = 'F'
                       AND strftime('%m', birth_date) IN ('2', '5', '12')
                       AND weight BETWEEN 60 AND 80
                       AND patient_id % 2 != 0
                       AND city = 'Kingston' ; ''')
```

49. What percent of our patients are male, rounded to the nearest whole number ?

```
cursor.execute(''' SELECT
                       ROUND(
                               (SUM(CASE WHEN gender = 'M' THEN 1 ELSE 0 END) * 100.0 / COUNT(*)),
                               2
                           ) || '%' AS percent_of_male_patients
                   FROM
                   patients; ''')
```

50. For each day display the total amount of admissions on that day. Display the amount changed from the previous date.

```
cursor.execute(''' SELECT admission_date,
                       COUNT(*) AS total_admissions,
                       COUNT(*) - LAG(COUNT(*)) OVER (ORDER BY admission_date) AS change_from_previous_day
                   FROM admissions
                   GROUP BY admission_date
                   ORDER BY admission_date; ''')
```

51. List all province names alphabetically, placing Ontario at the top.

```
cursor.execute(''' SELECT province_name
                   FROM province_names
                   ORDER BY
                       CASE
                           WHEN province_name = 'Ontario' THEN 0
                           ELSE 1
                       END,
                       province_name; ''')
```

52. We need a breakdown for the total amount of admissions each doctor has started each year. Show the doctor_id, doctor_full_name, specialty, year, total_admissions for that year.

```
cursor.execute(''' SELECT d.doctor_id, CONCAT(d.first_name, ' ', d.last_name) AS doctor_name, d.speciality,
                       strftime('%Y', a.admission_date) AS selected_year, COUNT(*) AS total_admissions
                   FROM doctors as d
                   LEFT JOIN admissions as a
                   ON d.doctor_id = a.attending_doctor_id
                   GROUP BY d.doctor_id, doctor_name, d.speciality, selected_year; ''')
```

Conclusion -

This comprehensive analysis of patient and admissions data will equip the hospital management team with actionable insights on patient demographics, admission patterns, and physician performance. By understanding patient distributions, identifying repeat admissions, and evaluating doctor engagement, the hospital can better allocate resources, improve care quality, and optimize operational efficiencies.