Medical Data History

**A hands-on SQL analytics project using a simulated medical database. It includes complex queries on patients, doctors, and admissions to extract insights, track trends, calculate BMI, and build reports using joins, aggregations, and conditional logic.**

**Perform the Problem Queries:**

1. Show first name, last name, and gender of patients who's gender is 'M'

**select first\_name,last\_name,gender from patients where gender='M';**

1. Show first name and last name of patients who does not have allergies.

**select first\_name,last\_name from patients where allergies is null;**

1. Show first name of patients that start with the letter 'C'

**select first\_name from patients where first\_name like 'c%';**

1. Show first name and last name of patients that weight within the range of 100 to 120 (inclusive)

**select first\_name,last\_name from patients where weight between 100 and 120;**

1. Update the patients table for the allergies column. If the patient's allergies is null then replace it with 'NKA'

**UPDATE patients**

**SET allergies = 'NKA'**

**WHERE allergies IS NULL;**

1. Show first name and last name concatenated into one column to show their full name.

**select concat(first\_name, ' ' ,last\_name ) as Full\_name from patients ;**

1. Show first name, last name, and the full province name of each patient.

**SELECT**

**p.first\_name,**

**p.last\_name,**

**pn.province\_name**

**FROM patients p**

**JOIN province\_names pn ON p.province\_id = pn.province\_id;**

1. Show how many patients have a birth\_date with 2010 as the birth year.

**select count(\*) from patients where year(birth\_date) = 2010;**

1. Show the first\_name, last\_name, and height of the patient with the greatest height.

**SELECT first\_name, last\_name, height**

**FROM patients**

**ORDER BY height DESC**

**LIMIT 1;**

1. Show all columns for patients who have one of the following patient\_ids: 1,45,534,879,1000

**select \* from patients where patient\_id in(1,45,534,879,1000);**

1. Show the total number of admissions

**select count(\*) as Total\_admissions from admissions;**

1. Show all the columns from admissions where the patient was admitted and discharged on the same day.

**SELECT \***

**FROM admissions**

**WHERE admission\_date = discharge\_date;**

1. Show the total number of admissions for patient\_id 579.

**select count(\*) as Total\_Admission from admissions where patient\_id = 579;**

1. Based on the cities that our patients live in, show unique cities that are in province\_id 'NS'?

**SELECT DISTINCT city**

**FROM patients**

**WHERE province\_id = 'NS';**

1. Write a query to find the first\_name, last name and birth date of patients who have height more than 160 and weight more than 70

**select first\_name, last\_name ,height, birth\_date from patients where height >70 order by height desc;**

1. Show unique birth years from patients and order them by ascending.

**SELECT DISTINCT YEAR(birth\_date) AS birth\_year**

**FROM patients**

**ORDER BY birth\_year ASC;**

17. Show unique first names from the patients table which only occurs once in the list.

**SELECT first\_name**

**FROM patients**

**GROUP BY first\_name**

**HAVING COUNT(\*) = 1;**

For example, if two or more people are named 'John' in the first\_name column then don't include their name in the output list. If only 1 person is named 'Leo' then include them in the output. Tip: HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

1. Show patient\_id and first\_name from patients where their first\_name start and ends with 's' and is at least 6 characters long.

**SELECT patient\_id, first\_name**

**FROM patients**

**WHERE first\_name LIKE 's%s'**

**AND LENGTH(first\_name) >= 6;**

1. Show patient\_id, first\_name, last\_name from patients whos diagnosis is 'Dementia'. Primary diagnosis is stored in the admissions table.

**SELECT p.patient\_id, p.first\_name, p.last\_name**

**FROM patients p**

**JOIN admissions a ON p.patient\_id = a.patient\_id**

**WHERE a.diagnosis = 'Dementia';**

1. Display every patient's first\_name. Order the list by the length of each name and then by alphbetically.

**SELECT first\_name, length(first\_name)**

**FROM patients**

**ORDER BY first\_name ASC;**

1. Show the total amount of male patients and the total amount of female patients in the patients table. Display the two results in the same row.

**SELECT**

**COUNT(CASE WHEN gender = 'M' THEN 1 END) AS male\_count,**

**COUNT(CASE WHEN gender = 'F' THEN 1 END) AS female\_count**

**FROM patients;**

1. Show the total amount of male patients and the total amount of female patients in the patients table. Display the two results in the same row.

**SELECT**

**COUNT(CASE WHEN gender = 'M' THEN 1 END) AS male\_count,**

**COUNT(CASE WHEN gender = 'F' THEN 1 END) AS female\_count**

**FROM patients;**

1. Show patient\_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis.

**SELECT patient\_id, diagnosis**

**FROM admissions**

**GROUP BY patient\_id, diagnosis**

**HAVING COUNT(\*) > 1;**

1. Show the city and the total number of patients in the city. Order from most to least patients and then by city name ascending.

**SELECT city, COUNT(\*) AS total\_patients**

**FROM patients**

**GROUP BY city**

**ORDER BY total\_patients DESC, city ASC;**

25. Show first name, last name and role of every person that is either patient or doctor. The roles are either "Patient" or "Doctor"

**SELECT first\_name, last\_name, 'Patient' AS role**

**FROM patients**

**UNION**

**SELECT first\_name, last\_name, 'Doctor' AS role**

**FROM doctors;**

26. Show all allergies ordered by popularity. Remove NULL values from query.

**SELECT allergies, COUNT(\*) AS frequency**

**FROM patients**

**WHERE allergies IS NOT NULL**

**GROUP BY allergies**

**ORDER BY frequency DESC;**

27. Show all patient's first\_name, last\_name, and birth\_date who were born in the 1970s decade. Sort the list starting from the earliest birth\_date.

**SELECT first\_name, last\_name, birth\_date**

**FROM patients**

**WHERE birth\_date BETWEEN '1970-01-01' AND '1979-12-31'**

**ORDER BY birth\_date ASC;**

28. We want to display each patient's full name in a single column. Their last\_name in all upper letters must appear first, then first\_name in all lower case letters. Separate the last\_name and first\_name with a comma. Order the list by the first\_name in decending order EX: SMITH,jane

**SELECT CONCAT(UPPER(last\_name), ',', LOWER(first\_name)) AS full\_name**

**FROM patients**

**ORDER BY first\_name DESC;**

29. Show the province\_id(s), sum of height; where the total sum of its patient's height is greater than or equal to 7,000.

**SELECT province\_id, SUM(height) AS total\_height**

**FROM patients**

**GROUP BY province\_id**

**HAVING SUM(height) >= 7000;**

30. Show the difference between the largest weight and smallest weight for patients with the last name 'Maroni'

**SELECT**

**MAX(weight) - MIN(weight) AS weight\_difference**

**FROM patients**

**WHERE last\_name = 'Maroni';**

31. Show all of 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.

**SELECT**

**DAY(admission\_date) AS day\_of\_month,**

**COUNT(\*) AS admission\_count**

**FROM admissions**

**GROUP BY day\_of\_month**

**ORDER BY admission\_count DESC;**

32. Show all of the patients grouped into weight groups. Show the total amount of patients in each weight group. Order the list by the weight group decending.

e.g. if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.

**SELECT**

**CASE**

**WHEN weight < 100 THEN 'Underweight'**

**WHEN weight BETWEEN 100 AND 149 THEN 'Normal'**

**WHEN weight BETWEEN 150 AND 199 THEN 'Overweight'**

**WHEN weight >= 200 THEN 'Obese'**

**ELSE 'Unknown'**

**END AS weight\_group,**

**COUNT(\*) AS patient\_count**

**FROM patients**

**GROUP BY weight\_group**

**ORDER BY weight\_group DESC;**

33. Show 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). Weight is in units kg. Height is in units cm.

**SELECT**

**patient\_id,**

**weight,**

**height,**

**CASE**

**WHEN weight / POWER(height / 100, 2) >= 30 THEN 1**

**ELSE 0**

**END AS isObese**

**FROM patients;**

34. Show patient\_id, first\_name, last\_name, and attending doctor's specialty. Show only the patients who has a diagnosis as 'Epilepsy' and the doctor's first name is 'Lisa'. Check patients, admissions, and doctors tables for required information.

**SELECT**

**p.patient\_id,**

**p.first\_name,**

**p.last\_name,**

**d.specialty**

**FROM patients p**

**JOIN admissions a ON p.patient\_id = a.patient\_id**

**JOIN doctors d ON a.attending\_doctor\_id = d.doctor\_id**

**WHERE a.diagnosis = 'Epilepsy'**

**AND d.first\_name = 'Lisa';**

35. 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

**SELECT**

**p.patient\_id,**

**CONCAT(**

**p.patient\_id,**

**LENGTH(p.last\_name),**

**YEAR(p.birth\_date)**

**) AS temp\_password**

**FROM patients p**

**WHERE p.patient\_id IN (**

**SELECT DISTINCT patient\_id FROM admissions**

**);**