



SQL
MEDICAL DATA HISTORY

TABLES

01

PATIENTS

02

DOCTORS

03

ADMISSIONS

PROJECT OVERVIEW

In this project, we ventured into the intricate world of medical data, armed with SQL as our tool of choice. Our mission? To extract meaningful insights from vast datasets, shedding light on crucial patterns and trends that can shape the future of healthcare. From patient demographics to disease prevalence, our analyses spanned a spectrum of topics, offering a comprehensive view of the healthcare landscape.



01

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

QUERY

```
select first_name, last_name, gender  
from patient  
where gender = 'M';
```

The screenshot shows the MySQL Workbench interface. At the top, there's a red bar with the title 'Result Grid' and various toolbar icons. Below the grid, there's a tab labeled 'patient 102'. The main area displays a table with three columns: 'first_name', 'last_name', and 'gender'. The data shows five rows of male patients: Donald Waterfield (M), Mickey Baasha (M), Jiji Sharma (M), Blair Diaz (M), and Charles Wolfe (M). In the bottom right corner of the main window, there's a message: '4530 row(s) returned'. Below the main window, there's a 'Output' section with an 'Action Output' tab. It lists two log entries: one for a previous query (SELECT patient_id, concat(patient_id, length(last_name), year(birth_date)) as temp_pas...) which returned 4530 rows, and another for the current query (select first_name, last_name, gender from patient where gender = 'M') which returned 2468 rows.

#	Time	Action	Message
142	09:26:40	SELECT patient_id, concat(patient_id, length(last_name), year(birth_date)) as temp_pas...	4530 row(s) returned
143	09:54:07	select first_name, last_name, gender from patient where gender = 'M'	2468 row(s) returned

› 2468 male patients present

02

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

QUERY

```
select first_name, last_name  
from patient  
where allergies is NULL;
```

The screenshot shows the MySQL Workbench interface with a query results grid and an action history panel.

Result Grid:

first_name	last_name
Donald	Waterfield
Blair	Diaz
Thomas	ONeill
Sonny	Beckett
Cedric	Coltrane

Action History:

#	Time	Action	Message
147	10:02:34	select first_name, last_name from patient where allergies = 'Sulfa'	157 row(s) returned
148	10:02:43	select first_name, last_name from patient where allergies = ''	2059 row(s) returned

> 2059 patients don't have any allergies

03

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

QUERY

```
select first_name  
from patient  
where first_name like 'C%';
```

04

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

QUERY

```
select first_name, last_name  
from patient  
where weight between 100 and 120;
```

05

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

QUERY

Update patient
set allergies = 'NKA'
Where allergies is null;

06

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

QUERY

select first_name, last_name,
concat(first_name, ' ', last_name) as full_name
from patient;

07

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

QUERY

```
select p.first_name, p.last_name, nm.province_name  
from patient p  
join province_names nm  
on p.province_id = nm.province_id;
```

08

Show how many patients have a birth_date with 2010 as the birth year.

QUERY

```
select count(*) as num_patients from patient  
where year(birth_date) = '2010';
```

09

Show the `first_name`, `last_name`, and `height` of the patient with the greatest height.

QUERY

```
select first_name, last_name, max(height) as greatest_height  
from patient  
group by first_name, last_name  
order by greatest_height desc  
limit 1;
```

10

Show all columns for patients who have one of the following `patient_ids`:
1,45,534,879,1000

QUERY

```
select * from patient  
where patient_id in (1,45,534,879,1000);
```

11

Show the total number of admissions

QUERY

```
select count(patient_id) as total_admissions from admission;
```

12

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

QUERY

```
select * from admission  
where admission_date = discharge_date;
```

13

Show the total number of admissions for patient_id 579.

QUERY

```
select count(patient_id) as total_admissions  
from admission  
where patient_id = 579;
```

14

Based on the cities that our patients live in, show unique cities that are in province_id NS ?

QUERY

```
select distinct(city) , province_id  
from patient  
where province_id = 'NS';
```

15

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

QUERY

```
select first_name, last_name, birth_date  
from patient  
where height > 160 and weight > 70;
```

16

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

QUERY

```
select distinct(year(birth_date)) as birth_year  
from patient  
order by birth_year asc;
```

17

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

QUERY

```
select first_name  
from patient  
group by first_name  
having count(first_name) = 1;
```

18

Show patient_id and first_name from patients where their first_name start and ends with s and is at least 6 characters long.

QUERY

```
select patient_id, first_name  
from patient  
where first_name like 's____%os';
```

19

Show patient_id, first_name, last_name from patients whos diagnosis is Dementia;. Primary diagnosis is stored in the admissions table.

QUERY

```
select a.patient_id, p.first_name, p.last_name  
from patient p  
join admission a  
on p.patient_id = a.patient_id  
where diagnosis = 'Dementia';
```

20

Display every patients first_name. Order the list by the length of each name and then by alphabetically.

QUERY

```
select first_name, length(first_name) as length  
from patient  
order by first_name asc;
```

21

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.

QUERY

```
select  
    sum(case when gender = 'M' then 1 else 0 end) as male_count,  
    sum(case when gender = 'F' then 1 else 0 end) as female_count  
from patient;
```

22

Show patient_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis.

QUERY

```
SELECT patient_id, diagnosis, COUNT(*) AS admission_count  
FROM admission  
GROUP BY patient_id, diagnosis  
HAVING COUNT(*) > 1;
```

23

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

QUERY

```
select city, count(patient_id) as num_patient  
from patient  
group by city  
order by num_patient asc ;
```

24

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

QUERY

```
SELECT first_name, last_name, 'Patient' AS role  
FROM patient  
UNION  
SELECT first_name, last_name, 'Doctor' AS role  
FROM doctor;
```

25

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

QUERY

```
SELECT first_name, last_name, 'Patient' AS role  
FROM patient  
UNION  
SELECT first_name, last_name, 'Doctor' AS role  
FROM doctor;
```

26

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

QUERY

```
select allergies, count(*) as popularity  
from patient  
where allergies is not null  
group by allergies  
order by popularity 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`.

QUERY

```
select first_name, last_name, birth_date  
from patient  
where year(birth_date) between 1970 and 1979  
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

QUERY

```
select concat(upper(last_name), ',', lower(first_name)) as full_name  
from patient  
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.

QUERY

```
select province_id, sum(height) as sum_height  
from patient  
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

QUERY

```
select (max(weight) - min(weight)) as difference  
from patient  
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.

QUERY

```
SELECT day(admission_date) AS day_num,  
       COUNT(*) AS num_of_admission  
  FROM admission  
 GROUP BY day_num  
 ORDER BY num_of_admission 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 descending. e.g. if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.

QUERY

```
SELECT  
       COUNT(*) AS patients_in_group,  
       FLOOR(weight / 10) * 10 AS weight_group  
  FROM patient  
 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.

Obesity is a medical condition characterized by an excessive accumulation of body fat, which poses a risk to health. It is typically defined using the Body Mass Index (BMI), a measure that relates weight to height. Here's a more detailed explanation:

Definition of Obesity:

Body Mass Index (BMI):

BMI Calculation: BMI is calculated by dividing a person's weight in kilograms by the square of their height in meters (kg/m^2).

BMI Categories:

Underweight: BMI less than 18.5

Normal weight: BMI 18.5–24.9

Overweight: BMI 25–29.9

Obesity: BMI 30 or higher

QUERY

```
SELECT patient_id, weight, height,  
CASE  
    WHEN weight/(POWER(height/100.0,2)) >= 30 THEN  
        1  
    ELSE  
        0  
    END) AS isObese  
FROM patient;
```

34

Show patient_id, first_name, last_name, and attending doctors specialty.
Show only the patients who has a diagnosis as Epilepsy; and the doctors first name is Lisa. Check patients, admissions, and doctors tables for required information.

QUERY

```
select p.patient_id as Patient_id,  
       p.first_name as Patient_first_name,  
       p.last_name as Patient_last_name,  
       d.specialty as attending_specialist  
  from patient p  
  join admission a  
    on a.patient_id = p.patient_id  
  join doctor d  
    on d.doctor_id = a.attending_doctor_id  
   where 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

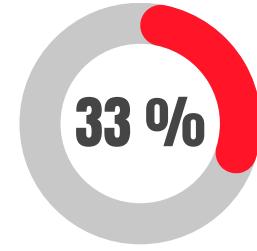
the numerical length of patients last_name

year of patients birth_date

QUERY

```
SELECT
    patient_id,
    concat(
        patient_id,
        length(last_name),
        year(birth_date)
    ) as temp_password
from
    patient
;
```

Findings



- **Gender Distribution of Patients:**
 - We identified and showcased the distribution of male patients within the dataset, providing insights into the gender demographics of healthcare recipients.
- **Patients without Allergies:**
 - By exploring patients without recorded allergies, we highlighted a subset of individuals who may have a different medical profile or require specific attention from healthcare providers.
- **Patient Demographics by Name:**
 - Extracting the first names of patients whose names start with a specific letter ('C') allowed us to delve into demographic patterns and trends within the dataset.
- **Weight Distribution Analysis:**
 - Investigating the weight distribution among patients within a specific range (100 to 120 pounds) provided insights into the health profiles of individuals falling within this weight bracket.
- **Enhanced Data Reliability:**
 - By updating null values in the allergies column with 'NKA' (No Known Allergies), we contributed to enhancing the reliability and completeness of the medical dataset.
- **Complex Aggregations and Relationships:**
 - Through intricate aggregations and table joins, we uncovered relationships within the dataset and tackled complex analyses, such as identifying patients with multiple admissions for the same diagnosis.
- **Continuous Learning and Real-world Applications:**
 - The project not only enhanced our proficiency in SQL and data analysis techniques but also underscored the importance of continuous learning and real-world applications of these skills in driving impactful data-driven decisions in healthcare.

Conclusion

Thank you for joining me on this enriching journey through the realms of SQL and medical data analysis. As we conclude this presentation, let us carry forth the insights gained, poised to make a positive impact on healthcare outcomes. Together, let us harness the transformative power of data analytics to drive innovation and change in the world of healthcare.

