❓ Why This Project ❓

What better feeling than to know that the work you accomplish can potentially save a life. My eagerness to become a data analyst in the healthcare industry has motivated me to take a journey through healthcare data and I am happy that I did, because I learned so much! I gathered insights that helped address the challenges confronting the hospital and improve the patient experience.

I will be your guide through this SQL safari and explain to you the queries that enabled me to obtain the insights requested by my imaginary higher ups in my local hospital! You will also understand how SQL queries can be used to solve real-world problems within the hospital! 🏥

🗄️ Data Source🗄️

This data was gathered from UC Irvine Machine Learning Repository and contains information from diabetes patients who received lab procedures, medication, and stayed for 14 days or more. The data was gathered from the year 1999 - 2008. This data is available to the public and I will include the link in the comments below so you can analyze the data too. 🔎

🔎 Analysis🔍

For this project, I have been hired by my local hospital as a data analyst. The Director of the hospital recognizes the power of data but struggles to interpret it effectively. There are many pressing tasks that the hospital would like me to address, so it is now my responsibility to advise the director and their superiors on data insights. 🕵️

⚕️ Task 1 ⚕️ : The hospital is experiencing overcrowding and needs to determine how to clear up beds.

To get a better understanding of how long patients are staying, the director wants to know the distribution of time spent in the hospital. I was able to deliver quick results to the director by implementing SQL's hidden visualization tool: THE HISTOGRAM!

```sql
-- This query creates the histogram needed to do some EDA and help determine the distribution of time at the hospital.
SELECT
    ROUND(time_in_hospital, 1) AS bucket,
    COUNT(*) AS count,
    RPAD('', COUNT(*)/100, '*') AS bar
FROM health
GROUP BY
    bucket
ORDER BY
    bucket;
```

The above query creates buckets that are used to group the patients by the number of days in the hospital. Each asterisk is a count of 200 patients. The asterisks then are stacked in each bucket, so the director can get a better understanding of how long patients are usually staying in the hospital. The query results are as shown below. 📝

The above query generates buckets that group patients by the duration of their stay in the hospital. Each asterisk represents 200 patients providing a visual representation of the typical hospital stays for the director's better understanding.

```sql
-- This query creates the histogram needed to do some EDA and help dete
SELECT
    ROUND(time_in_hospital, 1) AS days_in_hospital,
    COUNT(*) AS count,
    RPAD('', COUNT(*)/200, '*') AS bar
FROM health
GROUP BY
    days_in_hospital
ORDER BY
    days_in_hospital;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

days_in_hospital	count	bar
1	14208	**********
2	17224	**********
3	17756	**********
4	13924	**********
5	9966	*******
6	7539	******
7	5859	*****
8	4391	***
9	3002	**
10	2342	**
11	1855	*
12	1448	*
13	1210	*
14	1042	*

Based on the results of the histogram, patients commonly stay seven days or less. ✅

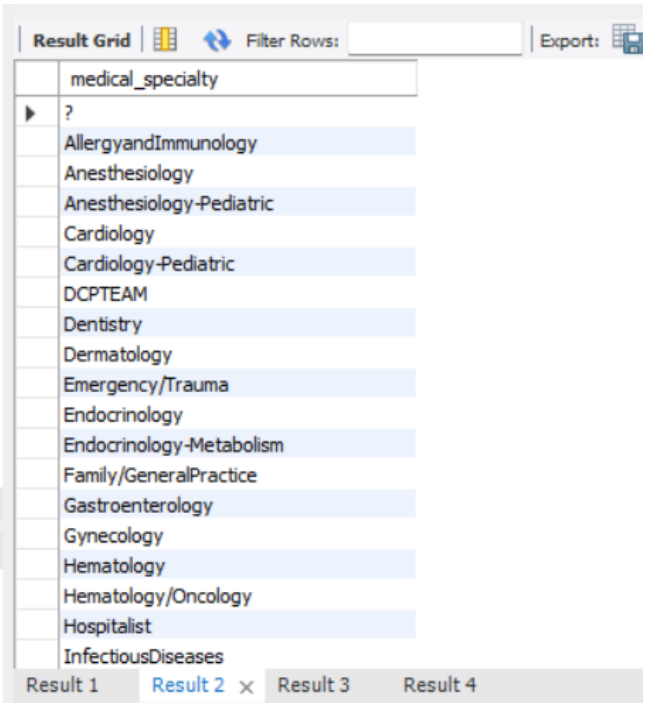
⚕ Task 2 ⚕ : One of the most significant costs incurred by the hospital is when specialists perform procedures. To determine how many procedures are being performed, the director asked us to create a list of the medical specialties within the hospital and their average count of total procedures completed.

This list will allow the director to hone a certain specialty's process & see any changes that need to be made. ↩

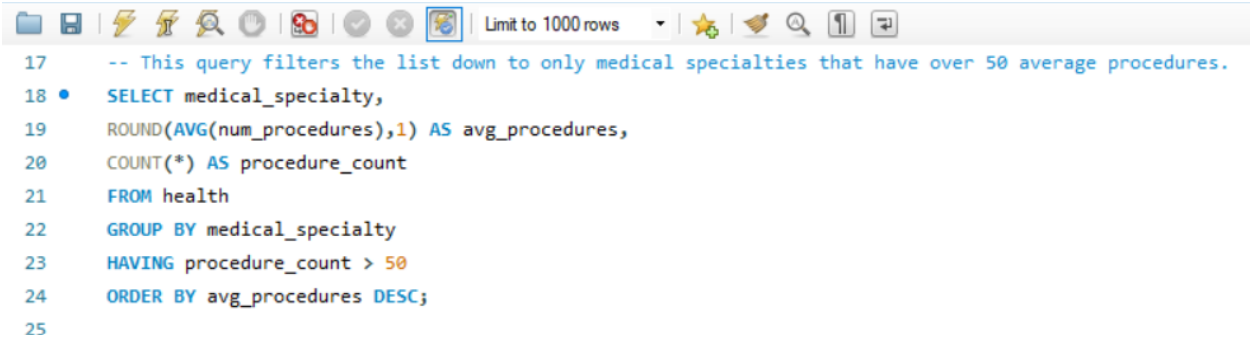
Limit to 1000 rows

```
12    -- To better understand the medical specialties within the data.
13 •  SELECT DISTINCT (medical_specialty)
14    FROM health
15    ORDER BY medical_specialty;
16
```

This query pulled the complete list of all the medical specialties in the hospital. It is always a clever idea to look over your data prior to trying to complete any kind of analysis. Below are the results of the above query. ⤵



Following my quick analysis of the list of medical specialties, I begin to filter the results down to what the director is looking for. Since they need a list of medical specialties and their average procedure count per patient, we will now work on getting that list.



To maximize readability, I used the ROUND function to remove many of the digits following the decimal (Shown in the results below). I used the HAVING clause to filter for medical specialties with an average procedure count over 50. Additionally, I ordered the results in descending order based on the average procedure count, highlighting the medical specialties with the highest average procedural count at the top of the list.

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

medical_specialty	avg_procedures	procedure_count
Surgery-Thoracic	3.5	109
Surgery-Cardiovascular/Thoracic	3.2	652
Radiologist	3.2	1140
Cardiology	2.7	5352
Surgery-Vascular	2.6	533
Radiology	2.5	53
Podiatry	2.4	100
Neurology	2.1	203
Gynecology	2.1	58
Surgery-Cardiovascular	2.1	98
Orthopedics	1.9	1400
Surgery-Neuro	1.9	468
ObstetricsandGynecology	1.9	671
Otolaryngology	1.9	125
Nephrology	1.8	1613
Urology	1.8	685
Orthopedics-Reconstructive	1.7	1233
Surgery-General	1.6	3099
Gastroenterology	1.6	564
Hematology	1.6	82
?	1.2	49949

As you can now see the average procedure count ranges from 3.5 to 1.2 in descending order. The director is now able to focus the attention on the medical specialties with the highest avg first, then analyze the remainder of the list.

To satisfy the director's urgent need for a response on the matter I will send him the above list. However, we are still able to create a more refined list to ensure clarity. 🖼️

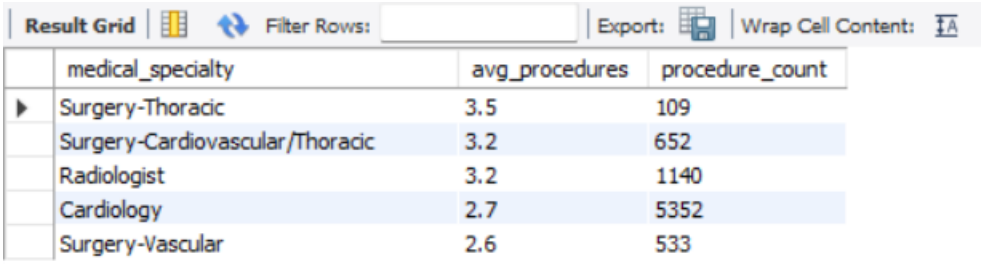
❓ (The question mark at the bottom of the list represents null values within the database that still need to be represented) ❓

Limit to 1000 rows

```sql
-- This query further filters the data down to a list of 5 medical specialties that have higher than a 2.5 avg procedures.
SELECT medical_specialty,
ROUND(AVG(num_procedures),1) AS avg_procedures,
COUNT(*) AS procedure_count
FROM health
GROUP BY medical_specialty
HAVING procedure_count > 50 AND avg_procedures > 2.5
ORDER BY avg_procedures DESC;
```

After receiving the previous list, the director specified that our focus is medical specialties with an average procedure per patient count of over 2.5. 💉🩺

With this new information, I added the following HAVING clause to further refine the list **(AND avg_procedures > 2.5)**.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

medical_specialty	avg_procedures	procedure_count
Surgery-Thoracic	3.5	109
Surgery-Cardiovascular/Thoracic	3.2	652
Radiologist	3.2	1140
Cardiology	2.7	5352
Surgery-Vascular	2.6	533

This list now provides the director with the medical specialties that require attention. Each of these performs over 2.5 average procedures. ☠️✅

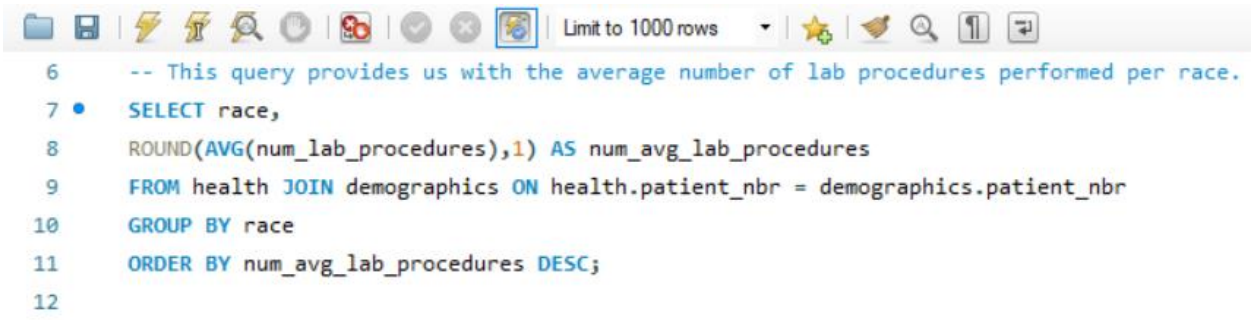
⚕️ Task 3 ⚕️ : The Chief Nursing Officer wants to examine any correlations between the number of procedures performed and the race of the patient. Their goal is to ensure racial discrimination does not occur within the hospital.

The two provided tables:

1) Demographics: Contains data about the patient's age, gender, ethnicity, etc... 🤚🏻🤚🏽🖐🏿

2) Health: Contains data regarding the person's health and visits to the hospital. 🧬🩸💊

To answer this question, we will need to use a JOIN clause to query both tables.
A JOIN clause is required to query both tables and answer this question. ➡️⬅️



Limit to 1000 rows

```
6   -- This query provides us with the average number of lab procedures performed per race.
7   SELECT race,
8   ROUND(AVG(num_lab_procedures),1) AS num_avg_lab_procedures
9   FROM health JOIN demographics ON health.patient_nbr = demographics.patient_nbr
10  GROUP BY race
11  ORDER BY num_avg_lab_procedures DESC;
12
```

This query will return the average number of lab procedures performed per race in a descending order as shown below. ⤵️

Result Grid | Filter Rows: | Export:

race	num_avg_lab_procedures
AfricanAmerican	44.1
?	44.0
Other	43.7
Caucasian	42.8
Hispanic	42.7
Asian	40.9

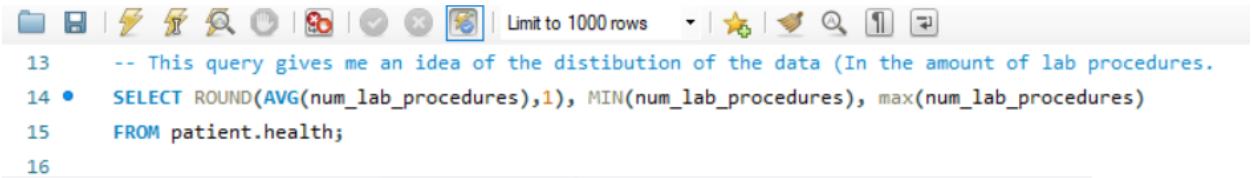
There is minimal difference in treatment between the races, which suggests no preferential treatment or discrimination.

The Chief Nursing Officer was pleased with the results but was interested in learning more.

⚕ Task 4 ⚕ : Having speculated a strong correlation between patient stay duration and the number of lab procedures performed, The Chief Nursing officer is now curious if the data will back this claim.

Before attempting to solve this problem, I will do some EDA and perform some aggregations.

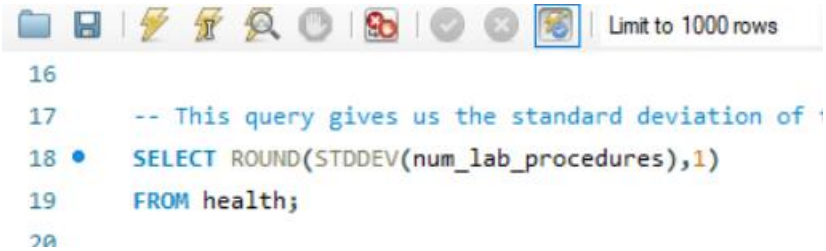
1️⃣ First, AVG, MIN, and MAX.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ROUND(AVG(num_lab_procedures),1)	MIN(num_lab_procedures)	max(num_lab_procedures)
43.1	1	132

2️⃣ Second is the standard deviation.



Result Grid | Filter Rows: | Exp

ROUND(STDDEV(num_lab_procedures),1)
19.7

After calculating the aggregate data, I began to bucket the number of lab procedures into groups of three: “many,” “average,” “few.” 🟦

```
21    -- This query gives us the buckets of how frequent the number of lab procedures are done.
22 •  SELECT num_lab_procedures,
23    CASE
24    WHEN num_lab_procedures >= 0 AND num_lab_procedures < 23 THEN "few"
25    WHEN num_lab_procedures >= 23 AND num_lab_procedures < 63 THEN "average"
26    ELSE "many"
27    END AS procedure_frequency
28    FROM patient.health
29    GROUP BY num_lab_procedures
30    ORDER BY procedure_frequency DESC;
31
```

I utilized CASE WHEN to categorize the number of lab procedures into three buckets. The standard deviation serves as the difference between the category minimums, determining when the transition will occur between stages. 😁

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

num_lab_procedures	procedure_frequency
109	many
104	many
96	many
106	many
108	many
107	many
70	many
73	many
68	many
129	many
111	many
113	many
114	many
101	many
99	many
75	many
92	many
91	many
89	many
100	many

Using the buckets created, I complete the request from the Chief Nursing Officer with the following query: ⤵️

```
31
32    -- This query is the final filtering of the data and provides us with the correlation betwe
33    SELECT
34    ROUND(AVG(time_in_hospital),1) AS days_in_hospital,
35    CASE
36    WHEN num_lab_procedures >= 0 AND num_lab_procedures < 23 THEN "few"
37    WHEN num_lab_procedures >= 23 AND num_lab_procedures < 63 THEN "average"
38    ELSE "many"
39    END AS procedure_frequency
40    FROM health
41    GROUP BY procedure_frequency
42    ORDER BY days_in_hospital DESC;
```

Result Grid | Filter Rows: | Expo

days_in_hospital	procedure_frequency
6.2	many
4.2	average
3.2	few

The result! ✅

We now see that the increase in patient stay duration and procedure count are positively correlated.

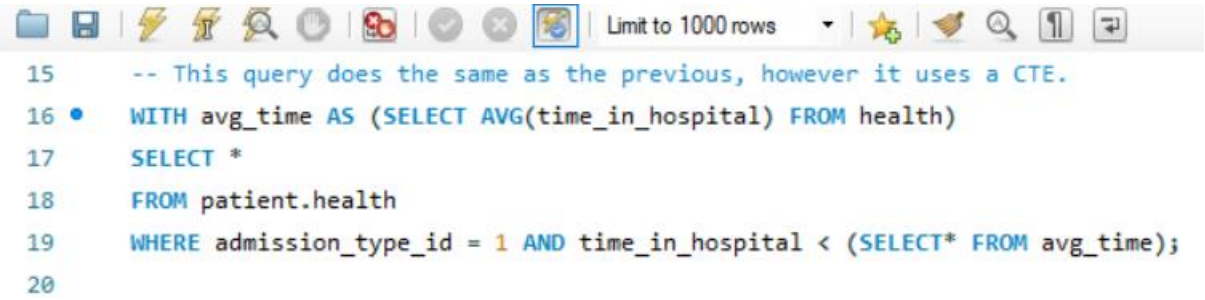
⚕️ Task 5 ⚕️ : The Hospital Administrator would like to highlight major successes within the hospital by viewing instances where patients were admitted into the hospital for an emergency and stayed less than average time.

I will start by filtering out the data with the necessary criteria: (admission type = 1 & less than average days stayed in hospital).

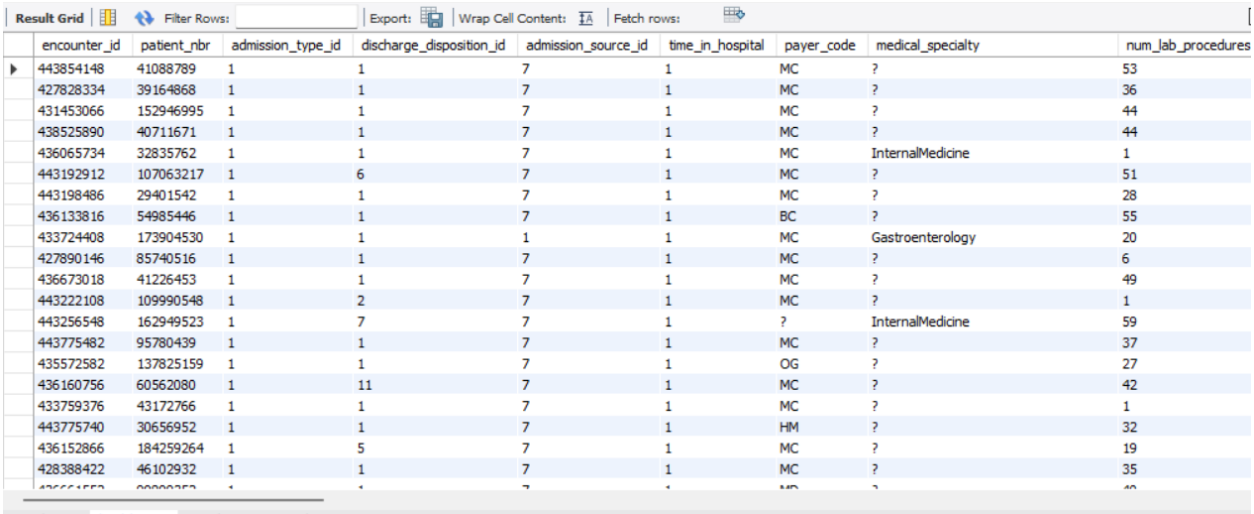
```
7     -- This query pulls patients that had an emergency (admission_type = 1)
8     SELECT *
9     FROM patient.health
10    WHERE admission_type_id = 1
11    AND time_in_hospital < (SELECT AVG(time_in_hospital)
12                            FROM patient.health)
13    ORDER BY time_in_hospital ASC;
```

In the above snippet I used a subquery in addition to the AND clause to include the additional filter of “time_in_hospital < AVG(time_in_hospital)” since MySQL does not allow for an AVG clause outside of a SELECT clause.

I retrieved the same data by utilizing a CTE (Common Table Expression).



And below are the results of the previous two queries. ⤵️



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

encounter_id	patient_nbr	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	payer_code	medical_specialty	num_lab_procedures
443854148	41088789	1	1	7	1	MC	?	53
427828334	39164868	1	1	7	1	MC	?	36
431453066	152946995	1	1	7	1	MC	?	44
438525890	40711671	1	1	7	1	MC	?	44
436065734	32835762	1	1	7	1	MC	InternalMedicine	1
443192912	107063217	1	6	7	1	MC	?	51
443198486	29401542	1	1	7	1	MC	?	28
436133816	54985446	1	1	7	1	BC	?	55
433724408	173904530	1	1	1	1	MC	Gastroenterology	20
427890146	85740516	1	1	7	1	MC	?	6
436673018	41226453	1	1	7	1	MC	?	49
443222108	109990548	1	2	7	1	MC	?	1
443256548	162949523	1	7	7	1	?	InternalMedicine	59
443775482	95780439	1	1	7	1	MC	?	37
435572582	137825159	1	1	7	1	OG	?	27
436160756	60562080	1	11	7	1	MC	?	42
433759376	43172766	1	1	7	1	MC	?	1
443775740	30656952	1	1	7	1	HM	?	32
436152866	184259264	1	5	7	1	MC	?	19
428388422	46102932	1	1	7	1	MC	?	35

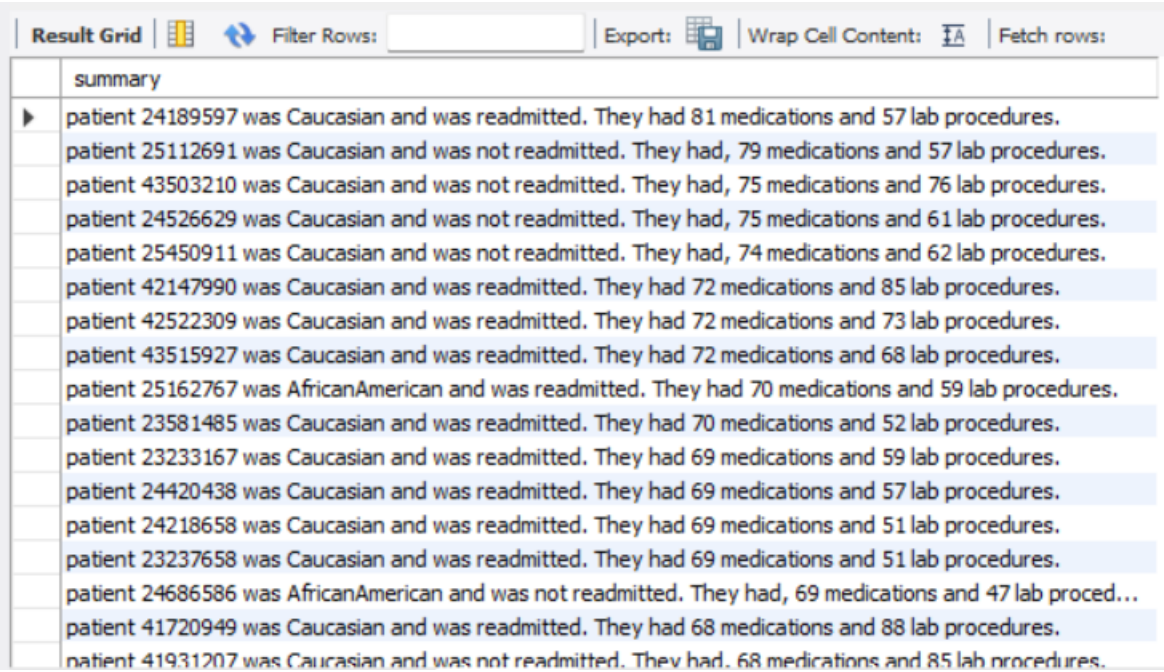
Now that The Hospital Administrator has reviewed the data, they have requested a summary of the top 50 patients by medication usage, along with the count of medications and lab procedures. The Administrator also prefers the data to be in sentence form for ease of reading. 🧐

I utilized the CONCAT clause to complete this task.

```sql
-- CONCAT function to form the sentence that states whether or not the patient was re
SELECT CONCAT_WS(' ', 'patient', health.patient_nbr, 'was', demographics.race, 'and',
(CASE WHEN readmitted = 'NO' THEN 'was not readmitted. They had,'
ELSE 'was readmitted. They had'
END),
num_medications, 'medications and', num_lab_procedures, 'lab procedures.')
AS summary FROM patient.health
INNER JOIN patient.demographics ON demographics.patient_nbr = health.patient_nbr
ORDER BY num_medications DESC, num_lab_procedures DESC
LIMIT 50;
```

The CONCAT clause combines two or more strings of data in MySQL. By adding the necessary columns to the clause, the data will be present in the resulting sentence. Additionally, filler words can be added to the resulting sentence by enclosing them in quotes.

And the results! 🥳🎉



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

summary
patient 24189597 was Caucasian and was readmitted. They had 81 medications and 57 lab procedures.
patient 25112691 was Caucasian and was not readmitted. They had, 79 medications and 57 lab procedures.
patient 43503210 was Caucasian and was not readmitted. They had, 75 medications and 76 lab procedures.
patient 24526629 was Caucasian and was not readmitted. They had, 75 medications and 61 lab procedures.
patient 25450911 was Caucasian and was not readmitted. They had, 74 medications and 62 lab procedures.
patient 42147990 was Caucasian and was readmitted. They had 72 medications and 85 lab procedures.
patient 42522309 was Caucasian and was readmitted. They had 72 medications and 73 lab procedures.
patient 43515927 was Caucasian and was readmitted. They had 72 medications and 68 lab procedures.
patient 25162767 was AfricanAmerican and was readmitted. They had 70 medications and 59 lab procedures.
patient 23581485 was Caucasian and was readmitted. They had 70 medications and 52 lab procedures.
patient 23233167 was Caucasian and was readmitted. They had 69 medications and 59 lab procedures.
patient 24420438 was Caucasian and was readmitted. They had 69 medications and 57 lab procedures.
patient 24218658 was Caucasian and was readmitted. They had 69 medications and 51 lab procedures.
patient 23237658 was Caucasian and was readmitted. They had 69 medications and 51 lab procedures.
patient 24686586 was AfricanAmerican and was not readmitted. They had, 69 medications and 47 lab proced...
patient 41720949 was Caucasian and was readmitted. They had 68 medications and 88 lab procedures.
patient 41931207 was Caucasian and was not readmitted. They had, 68 medications and 85 lab procedures.

Thankfully with time and dedication I was able to handle all the requests from my higher ups! The Director was incredibly happy with my performance, and we will continue to make data-driven decisions that save patient lives and improve the patient's experience. 💥

Now that you have finished reading this article, you have gained a better understanding of how SQL functions can be used to solve real world problems in the hospital. ✅

This project, although a fictional scenario, has been very insightful into the world of healthcare analytics. I am excited to begin exploring opportunities within the healthcare

industry as a data analyst. If you have any additional insights you want to add, please comment on this post and we can discuss! If you are a hiring manager or recruiter and need a data analyst on your team, please send me a direct message and I would love to explain why I am a great fit for your organization. 🤝 📩