

QUERYING FOR CARE

**THE EPIC QUEST TO
CONQUER DEPTHS OF
HOSPITAL STAYS
ONE QUERY AT A TIME!**



By Quy Tran

Have you ever found yourself anxiously waiting in a hospital? You're not alone! This stressful experience inspired me to unmask the Secrets of Hospital Operation. Using MySQL magic 🧙.



Why THIS Project?

- **Belief in Effective Healthcare:** Good health is vital for a fulfilling life.
- **Addressing Challenges:** Overwhelming medical expenses and long hospital stays are daunting.
- **Goal:** Optimize hospital operations to provide exceptional care without needing more resources.



What Readers Will Gain

- Discover insights from analyzing hospital operations using SQL queries.
- Understand patient trends, medical specialties, and medication usage.
- Learn the importance of data-driven insights in enhancing patient care.



Key Takeaways

- **Hospital Stays:** Brief stays—Most patients stay between 2-3 days.
- **Medical Specialties:** Leading fields are thoracic surgery, radiology, and cardiology.
- **Patient Demographics:** Caucasian patients undergo the most procedures and require the highest medication usage.
- **Emergency Admission:** Majority of patients had under one emergency



Dataset Details

- **Source:** Center for Clinical and Translational Research at Virginia Commonwealth University.
- **Scope:** A decade of data from 130 US hospitals focusing on diabetic inpatient encounters.
- **Features:** Over 50 patient and hospital outcome features for comprehensive trend analysis.

Analysis Process

1. Data Preparation:

- Cleaned data and understood various attributes.

Method:

MySQL WorkBench method:

The screenshot shows the MySQL WorkBench interface with the 'Schemas' tab selected. Under the 'Information' section, the 'Table: demographics' is highlighted. The 'Columns:' section lists the following columns:

Column Name	Data Type
patient_nbr	varchar(255)
race	varchar(255)
gender	varchar(255)
age	varchar(255)
weight	varchar(255)
rowNumber	int
RID	bigint UN

MySQL command:

```
"Check data type"  
USE patient;  
SELECT column_name, DATA_TYPE  
FROM INFORMATION_SCHEMA.COLUMNS  
WHERE table_schema = 'patient' and table_name = 'demographics';
```

Result:

Demographics Table:

COLUMN_NAME	DATA_TYPE
age	varchar
gender	varchar
patient_nbr	varchar
race	varchar
RID	bigint
rowNumber	int
weight	varchar

Health Table:

Table: health

Columns:

encounter_id	varchar(255)
patient_nbr	varchar(255)
admission_type_id	varchar(255)
discharge_disposition_id	varchar(255)
admission_source_id	varchar(255)
time_in_hospital	varchar(255)
payer_code	varchar(255)
medical_specialty	varchar(255)
num_lab_procedures	varchar(255)
num_procedures	varchar(255)
num_medications	varchar(255)
number_outpatient	varchar(255)
number_emergency	varchar(255)
number_inpatient	varchar(255)
diag_1	varchar(255)
diag_2	varchar(255)
diag_3	varchar(255)
number_diagnoses	varchar(255)
max_glu_serum	varchar(255)
A1Cresult	varchar(255)
metformin	varchar(255)
repaglinide	varchar(255)
nateglinide	varchar(255)
chlorpropamide	varchar(255)
glimepiride	varchar(255)
acetohexamide	varchar(255)
glipizide	varchar(255)
glyburide	varchar(255)
tolbutamide	varchar(255)
pioglitazone	varchar(255)
rosiglitazone	varchar(255)
acarbose	varchar(255)
miglitol	varchar(255)
troglitazone	varchar(255)
tolazamide	varchar(255)
examide	varchar(255)
citoglipiton	varchar(255)
insulin	varchar(255)
glyburide_metformin	varchar(255)
glipizide_metformin	varchar(255)
glimepiride_pioglitazone	varchar(255)
metformin_rosiglitazone	varchar(255)
metformin_pioglitazone	varchar(255)
change1	varchar(255)
diabetesMed	varchar(255)
readmitted	varchar(255)

b. Correct data type:

Code:

```
"Correct Data Type"  
ALTER TABLE health MODIFY COLUMN num_medications INT;  
ALTER TABLE health MODIFY COLUMN num_lab_procedures INT;
```

Result:

Table: health

Columns:

encounter_id	varchar(255)
patient_nbr	int
admission_type_id	varchar(255)
discharge_disposition_id	varchar(255)
admission_source_id	varchar(255)
time_in_hospital	varchar(255)
payer_code	varchar(255)
medical_specialty	varchar(255)
num_lab_procedures	int
num_procedures	int
num_medications	int
number_outpatient	int
number_emergency	int
number_inpatient	int

c. Verify patient_nbr is distinct:

Code

```
"Double check there is no duplicate patient_nbr"  
Method 1:  
SELECT COUNT(patient_nbr) AS count, COUNT(DISTINCT(patient_nbr)) AS dist_count FROM demographics;  
Method 2:  
SELECT patient_nbr AS nbr, COUNT(patient_nbr) AS count FROM demographics  
GROUP BY nbr  
ORDER BY count;
```

Result:

Method 1:

count	dist_count
71518	71518

-> good

Method 2:

nbr	count
98884206	1
98884809	1
98886357	1
98887572	1

d. Double check that there not multiple demographics for the same patient

Code

```
SELECT health.patient_nbr AS nbr, COUNT(DISTINCT(demographics.race)) AS race_cnt FROM health  
JOIN demographics ON demographics.patient_nbr = health.patient_nbr  
GROUP BY nbr  
ORDER BY race_cnt DESC;
```

Result:

nbr	race_cnt
377964	1
377712	1
377523	1
376947	1
376398	1
376200	1
375921	1
375642	1

2. Histogram

a. Time in Hospital Histogram:

- Most patients stay less than 7 days, with peaks around 2-3 days.
- Encouraging sign of effective short-term care.

Code:

```
"Time_In_Hospital_Histogram"
SELECT ROUND(CAST(time_in_hospital AS DECIMAL), 1) AS bucket,
       COUNT(*) AS count,
       RPAD(' ', COUNT(*)/150, '*') AS bar
  FROM patient.health
 GROUP BY bucket
 ORDER BY bucket;
```

Result:

bucket	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	*****

b. Emergency Admissions Histogram:

- Majority had under one admission; few faced multiple emergencies.
- Indicates a need for targeted prevention strategies.

Code:

```
"Number_of_Emergency_Histogram"
SELECT ROUND(CAST(number_emergency AS DECIMAL), 1) AS bucket,
       COUNT(*) AS count,
       RPAD(' ', COUNT(*)/150, '*') AS bar
  FROM patient.health
 GROUP BY bucket
 ORDER BY bucket;
```

Result:

bucket	count	bar
0	90383	*****
1	7677	*****
2	2042	*****
3	725	****
4	374	**
5	192	*
6	94	*

Histogram Insights:

 Most patients have brief stays (2-3 days).

 Understanding patient flow helps hospitals manage resources effectively.

3. Procedures by Specialty:

- Highest average procedures in thoracic and cardiovascular surgeries, radiology, and cardiology.
- Highlights areas for resource allocation.

Code:

```
List all medical specialty with at least 50 patients with at least 2.5 average procedures:  
SELECT medical_specialty, ROUND (AVG(num_procedures),1) AS avg_procedures, COUNT(*) AS count FROM patient.health  
GROUP BY medical_specialty  
HAVING count >50 AND avg_procedures >2.5  
ORDER BY avg_procedures DESC;
```

Result:

medical_specialty	avg_procedures	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

 Look like matter of the heart is in the lead.

4. Total number of procedures by medical specialty:

That's great. However, I want to know in total, number of procedures the hospital has to prepare for a specialty.

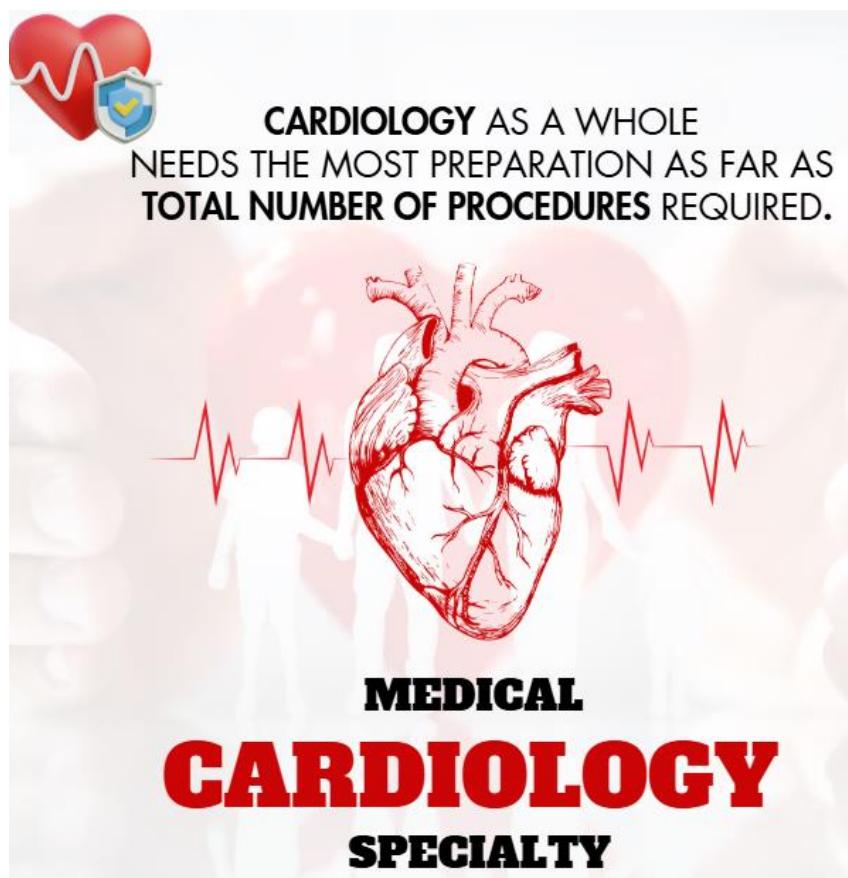
Let create a column to measure the product of avg procedures and number of patients for medical specialty:

Code:

```
Create a column to measure the product of avg procedures and number of patients for medical specialty:  
SELECT medical_specialty,  
ROUND(AVG(num_procedures),1) AS avg_procedures,  
COUNT(*) AS count,  
ROUND(COUNT(*) * AVG(num_procedures), 1) AS avg_procedures_prod  
FROM patient.health  
GROUP BY medical_specialty  
HAVING count >20 AND avg_procedures >1  
ORDER BY avg_procedures_prod DESC;
```

Result:

medical_specialty	avg_procedures	count	avg_procedures_prod
?	1.2	49949	61821
Cardiology	2.7	5352	14474
Emergency/Trauma	1.2	7565	8732
Surgery-General	1.6	3099	5110
Radiologist	3.2	1140	3696
Nephrology	1.8	1613	2872
Orthopedics	1.9	1400	2691
Surgery-Cardiovascular/Thoracic	3.2	652	2116
Orthopedics-Reconstructive	1.7	1233	2083
Surgery-Vascular	2.6	533	1370
ObstetricsandGynecology	1.9	671	1294
Urology	1.8	685	1253
Surgery-Neuro	1.9	468	908
Gastroenterology	1.6	564	880
Neurology	2.1	203	420
Oncology	1.1	348	382
Surgery-Thoracic	3.5	109	382
Podiatry	2.4	100	240
Otolaryngology	1.9	125	238
Surgery-Cardiovascular	2.1	98	207
Radiology	2.5	53	134
Hematology	1.6	82	129
Gynecology	2.1	58	121
Surgeon	2.2	45	100
Surgery-Plastic	2.3	41	94
Obsterics&Gynecology-Gynecol...	3.6	25	90
Ophthalmology	1.5	38	57
InfectiousDiseases	1.2	37	45



5. Pediatrics Procedures

Create a list of Pediatrics medical_specialty name as medical_clean

Code:

```
USE patient;
• SELECT
  Ⓜ CASE
    WHEN medical_specialty LIKE '%Pediatric%' THEN 'Pediatric'
    ELSE medical_specialty
  END AS medical_clean,
  medical_specialty
  FROM health
  WHERE medical_specialty LIKE '%PEDIATRICS%'
  ORDER BY medical_specialty DESC;
```

Result:

medical_clean	medical_specialty
Pediatric	Pediatrics-Pulmonology
Pediatric	Pediatrics-Neurology
Pediatric	Pediatrics-InfectiousDiseases
Pediatric	Pediatrics-Hematology-Oncology
Pediatric	Pediatrics-Endocrinology
Pediatric	Pediatrics-Endocrinology

6. Lab procedures by race – (INNER) JOIN (SIDE BY SIDE – ADDING COLUMN)

Code:

```
Race vs number of procedures:
SELECT race, ROUND (AVG(num_procedures),1) AS avg_procedures, COUNT(*) AS count
FROM health JOIN demographics ON demographics.patient_nbr = health.patient_nbr
GROUP BY race
ORDER BY avg_procedures DESC;
```

Result:

race	avg_procedures	count
Other	1.5	1524
Caucasian	1.4	76107
Asian	1.3	638
?	1.3	2280
AfricanAmerican	1.2	19198
Hispanic	1.1	2019

Querying for Care

LAB PROCEDURE BY RACE

CAUCASIAN

- Highest average number of procedure
- largest group who have procedure

ASIAN

Second highest average number of procedure

7. Correlate between the number of lab procedures and the number of days in hospital
- a. Understand number of lab procedures

Code:

```
"Correlate between the number of lab procedures and to the number of days in hospital"  
"Quick view at num_lab_procedures attribute"  
SELECT MIN(num_lab_procedures), ROUND(AVG(num_lab_procedures),2), MAX(num_lab_procedures) FROM health;
```

Result:

MIN(num_lab_procedures)	ROUND(AVG(num_lab_procedures),2)	MAX(num_lab_procedures)
1	43.1	99

- ☞ Average number of lab procedures is 43.1
- ☞ Most patients have average number of procedures ranking between 25 to 55
- ☞ Average time in the hospital of 4.1.

b. Creating “procedure_frequency”

Code:

```
"Create procedure_frequency column."
SELECT
    COUNT(*) AS count,
    ROUND(AVG(time_in_hospital),1) AS avg_time,
    CASE
        WHEN num_lab_procedures>=0 AND num_lab_procedures <25 THEN "few"
        WHEN num_lab_procedures>=25 AND num_lab_procedures <55 THEN "average"
        ELSE "many" END AS procedure_frequency
FROM patient.health
GROUP BY procedure_frequency
ORDER BY avg_time DESC;
```

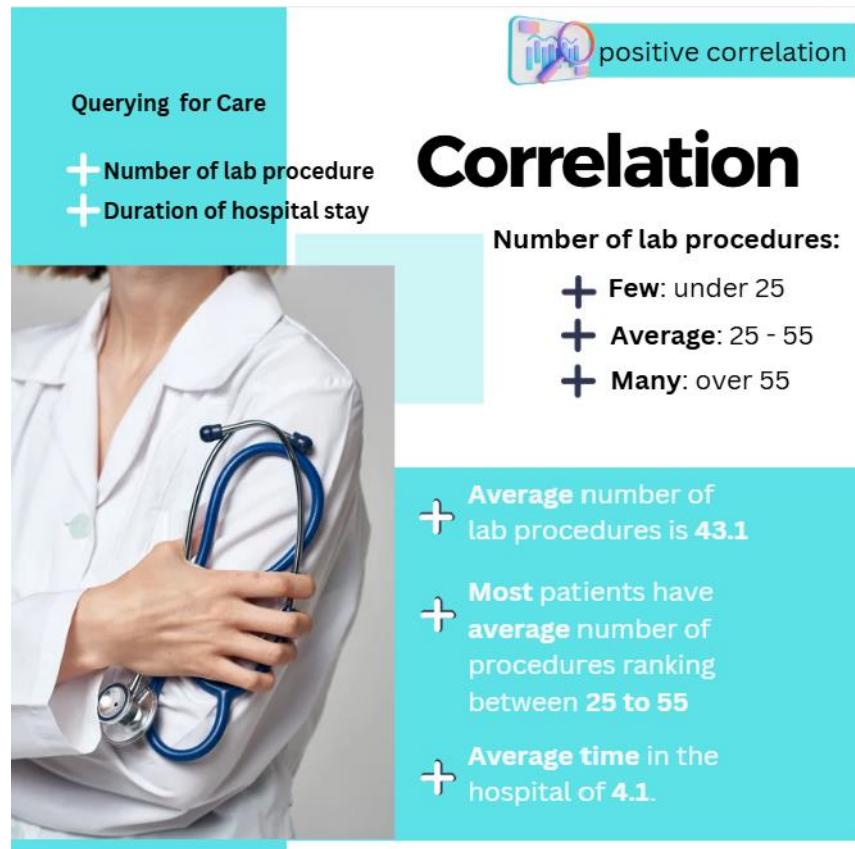
Result:

count	avg_time	procedure_frequency
29779	5.7	many
54501	4.1	average
17486	3.3	few

💡 Higher procedure frequency stays longer in the hospital.

⌚ Unsure the cause and effect of this relationship.

Whether patients stay longer due to complexity of their condition hence require more lab procedures, or due to the time requires to perform these lab procedures they must have.



8. African American Patient with metformin

List of Patient ID for Up metformin for African American – UNION (Stack-Adding rows)

Code:

```
"List of Patient ID for Up metformin for African American"
SELECT patient_nbr FROM demographics WHERE race = 'AfricanAmerican'
UNION SELECT patient_nbr FROM health WHERE metformin = 'Up';
```

Result:

patient_nbr
100000170
100002906
10002096
100022508
100041714
100042416
1000494
100062702
100070019
100071756
100073052
100091817
100109826
100119267
100141308

Result 60 ×

9. Highlight some of the biggest success stories of the hospital:

Filter:

- Emergency admission_type_id of 1
- Stayed less than the average time in the hospital

Sub-Query vs CTE (Common Table Expressions) method

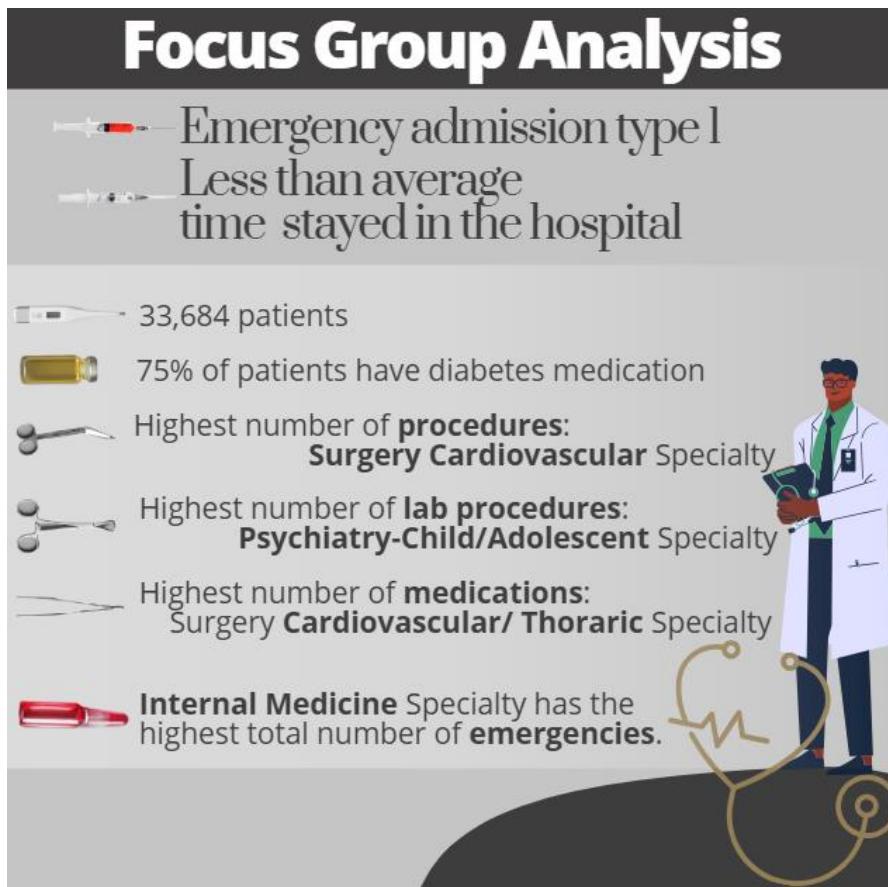
Code:

```
"Highlight some of the biggest success stories of the hospital;
Emergency_admission_type_id of 1 stayed less than the average time in the hospital."
SUBQUERY METHOD:
SELECT * FROM patient.health
WHERE admission_type_id = 1
AND time_in_hospital < (SELECT AVG (time_in_hospital) FROM health);

CTE METHOD:
WITH avg_time AS (SELECT AVG (time_in_hospital) FROM health)
SELECT * FROM patient.health
WHERE admission_type_id = 1
AND time_in_hospital < (SELECT * FROM avg_time);
```

Result:

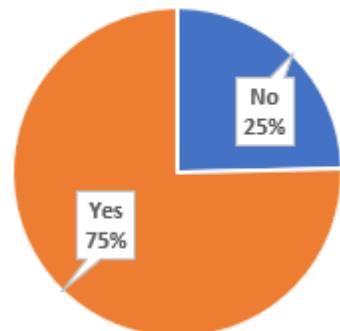
encounter_id	patient_nbr	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	payer_code	medical_specialty	num_lab_procedures	num_procedures	num_medications	number_outpatient	number_emergency	number_inpatient	diag_1	diag_2	diag_3	number_diagnoses
149190	55629189	1	1	7	3	?	?	59	0	18	0	0	0	276	250.01	255	9
64410	86047075	1	1	7	2	?	?	11	5	13	2	0	1	648	250	V27	6
500364	82442376	1	1	7	2	?	?	44	1	16	0	0	0	8	250.43	403	7
16680	42519267	1	1	7	1	?	?	51	0	8	0	0	0	197	157	250	5
77076	92519352	1	1	7	4	?	?	45	4	17	0	0	0	410	411	414	8
84222	108662661	1	1	7	3	?	?	29	0	11	0	0	0	682	174	250	3
221634	21861756	1	1	7	1	?	?	33	0	7	0	0	0	786	401	250	3
248916	115196778	1	1	1	2	?	Surgery-General	25	2	11	0	0	0	996	585	250.01	3
253722	96664626	1	5	7	1	?	?	53	0	10	0	0	0	462	250.01	276	8
325848	63023292	1	1	7	2	?	Cardiology	41	0	11	0	0	0	411	250.01	401	6
326028	112002975	1	1	7	4	?	?	33	0	12	0	0	0	486	244	250	3
383430	80588529	1	2	7	1	?	Cardiology	28	0	15	0	0	0	414	411	250.01	4
449142	66274866	1	1	7	3	?	Family/GeneralP...	59	0	11	0	0	0	428	496	278	6
591996	58497651	1	1	7	1	?	?	53	0	11	1	0	0	786	427	278	9



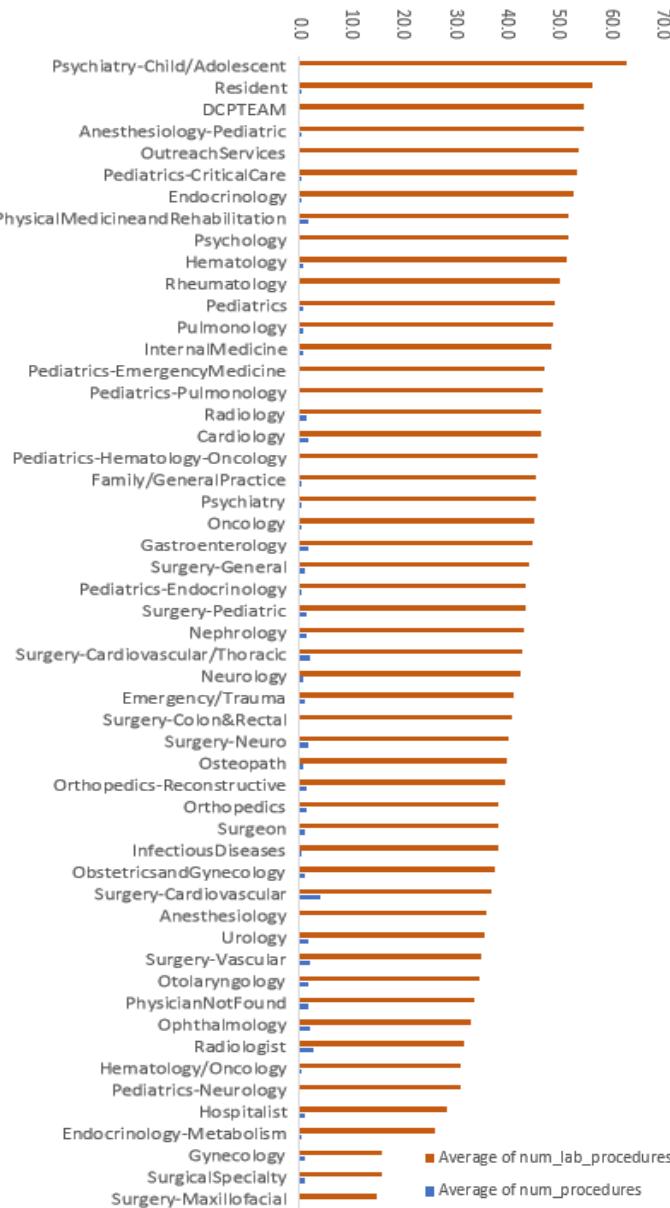
Number of Patients

33,684

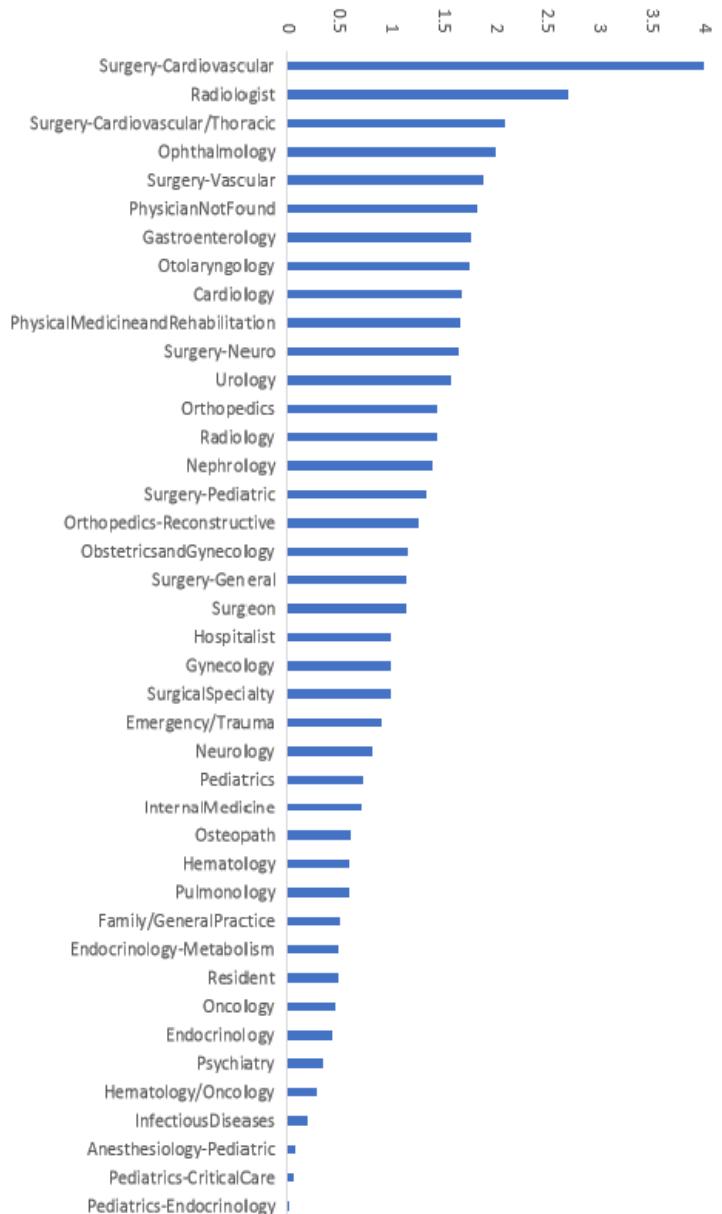
Diabetes Med



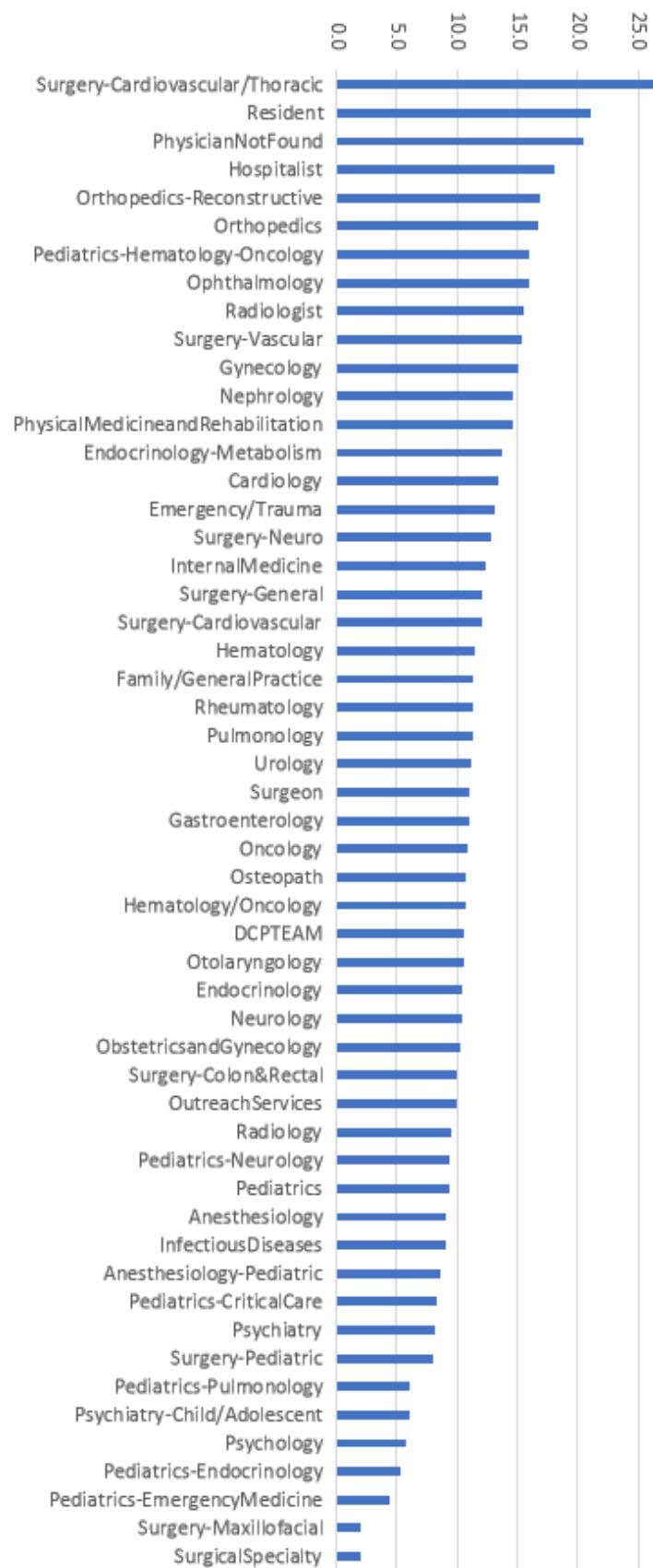
Average of num_lab_procedures and num_procedures by Medical Specialties



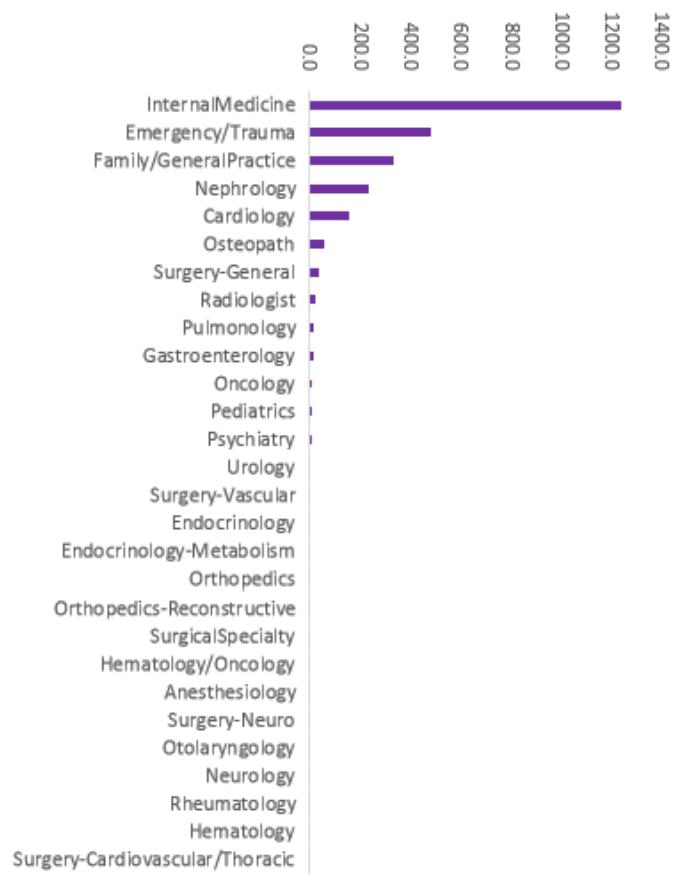
Average of num_procedures by Medical Specialties



Average of num_medications by Medical Specialties



Total of number of emergencies by Medical Specialties



10. Top 50 medication patients - CONCAT (concatenation)

Write a summary for the top 50 medication patients, break any ties with the number of lab procedures (highest at the top) in the following format:

Code:

```
"Top_50_medication_patients"
SELECT CONCAT('Patient ', health.patient_nbr, ' was a ', demographics.age, ' years old ',
    | demographics.gender, ' ', 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;
```

Result:

summary
Patient 24189597 was a [60-70)years old Male Caucasian and was readmitted. They had 81 medications and 57 lab procedures.
Patient 25112691 was a [50-60)years old Male Caucasian and was not readmitted. They had 79 medications and 57 lab procedures.
Patient 43503210 was a [60-70)years old Male Caucasian and was not readmitted. They had 75 medications and 76 lab procedures.
Patient 24526629 was a [60-70)years old Female Caucasian and was not readmitted. They had 75 medications and 61 lab procedures.
Patient 25450911 was a [70-80)years old Male Caucasian and was not readmitted. They had 74 medications and 62 lab procedures.
Patient 42147990 was a [70-80)years old Male Caucasian and was readmitted. They had 72 medications and 85 lab procedures.
Patient 42522309 was a [70-80)years old Male Caucasian and was readmitted. They had 72 medications and 73 lab procedures.
Patient 43515927 was a [60-70)years old Male Caucasian and was readmitted. They had 72 medications and 68 lab procedures.

📌 Main Takeaways



Querying for Care

Main Takeaways

- ✓ Brief stays: Most patients stay between 2-3 days
- ✓ Emergency Admission: Majority had under one
- ✓ Departments: cardiology and radiology
- ✓ Demographics: male Caucasian
- ✓ Higher procedure frequency stays longer

[SQL Project](#)



使命感 🧠 Conclusion and Personal Reflections

LEARNING JOURNEY +

Navigating complex data was **challenging**, but creating simpler visuals made **insights** accessible.

WE CAN DO IT.

🌐 <https://qtt4423.github.io/>
🌐 <https://www.linkedin.com>



Commitment

Reinforced my dedication to using analytics for positive changes in patient care.

🌐 Querying for Care
🌐 https://qtt4423.github.io/
👤 Quy Tran



🤝 Call to Action

- **Connect with Me:** Reach out on LinkedIn if you're interested in data analysis or know someone looking to hire a data analyst.
- **Let's Talk:** Join the conversation on enhancing healthcare through the power of data!