SQL Project

Massachusetts General Hospital Recent Performance

Presented By:

MAHESH Y

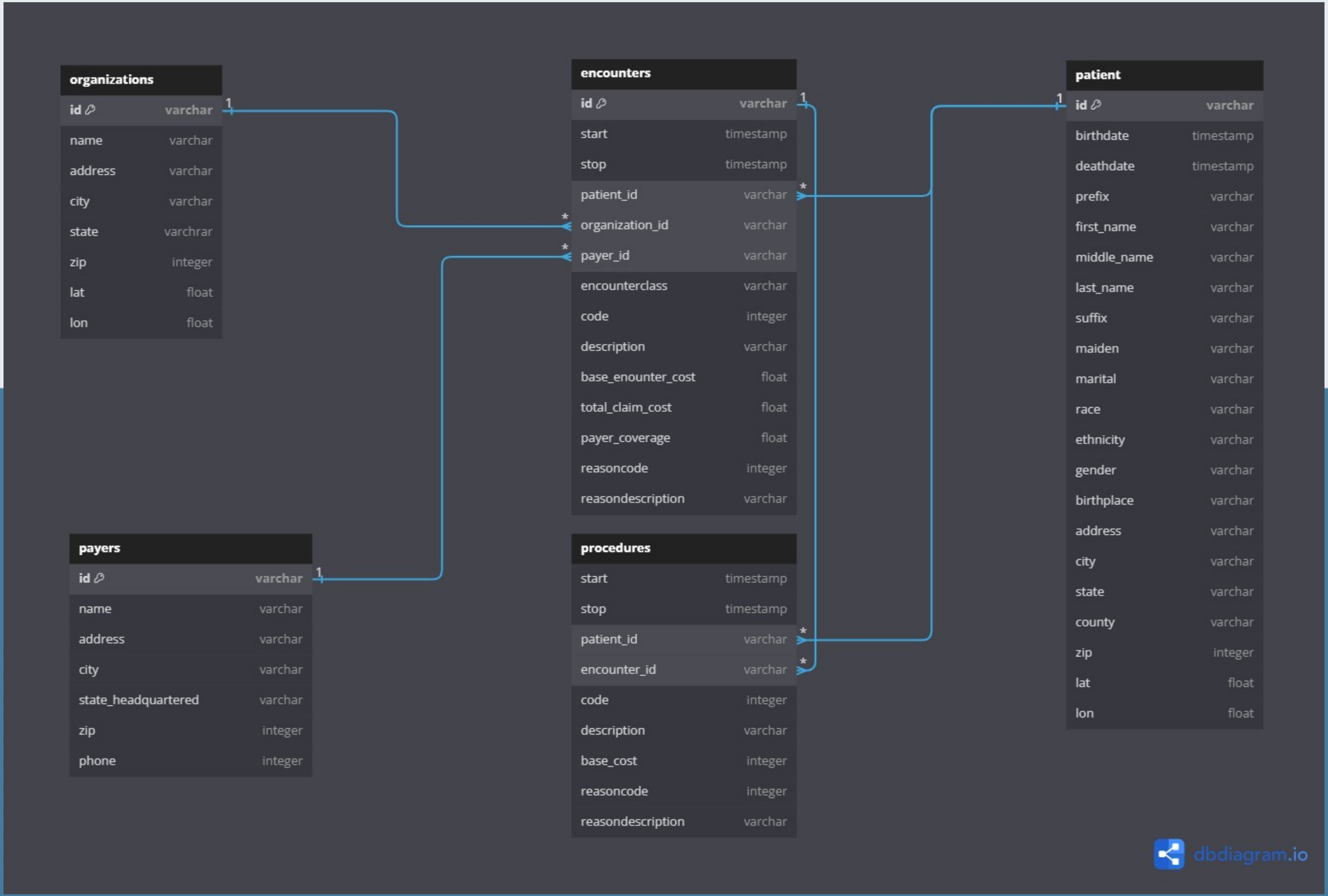
Tools:
MYSQL – QUERIES | DBDIAGRAM.IO – ERD | DATAWRAPPER - VISUALS

SOURCE: Maven Analytics

DATA:
Synthetic data on ~1k patients of Massachusetts General Hospital from 2011-2022, including information on patient demographics, insurance coverage, and medical encounters & procedures.

The Hospital Dataset (HSP) is a comprehensive collection of patient encounter data, spanning 10 years. It contains valuable information on patient demographics, encounter descriptions, and associated metrics such as patient counts and encounter frequencies.

Database Schema:



organizations	
id	varchar
name	varchar
address	varchar
city	varchar
state	varchrar
zip	integer
lat	float
lon	float

encounters	
id	varchar
start	timestamp
stop	timestamp
patient_id	varchar
organization_id	varchar
payer_id	varchar
encounterclass	varchar
code	integer
description	varchar
base_enounter_cost	float
total_claim_cost	float
payer_coverage	float
reasoncode	integer
reasondescription	varchar

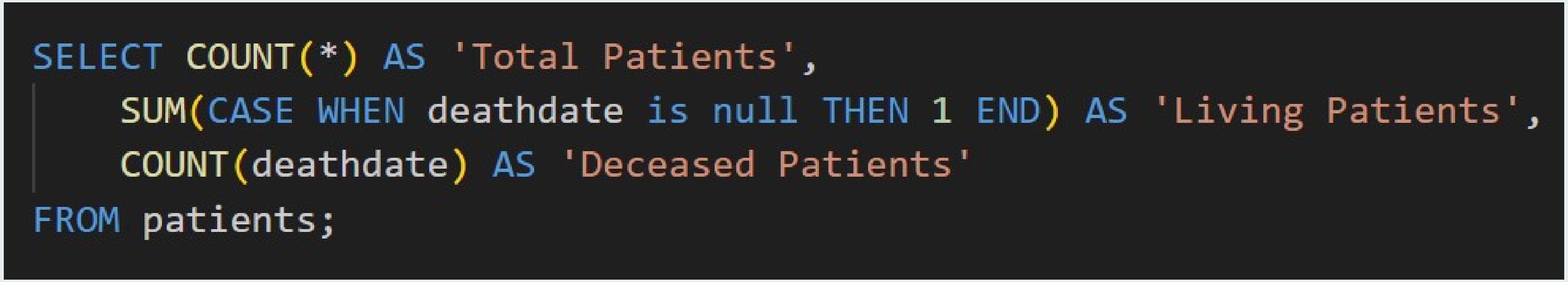
patient	
id	varchar
birthdate	timestamp
deathdate	timestamp
prefix	varchar
first_name	varchar
middle_name	varchar
last_name	varchar
suffix	varchar
maiden	varchar
marital	varchar
race	varchar
ethnicity	varchar
gender	varchar
birthplace	varchar
address	varchar
city	varchar
state	varchar
county	varchar
zip	integer
lat	float
lon	float

payers	
id	varchar
name	varchar
address	varchar
city	varchar
state_headquartered	varchar
zip	integer
phone	integer

procedures	
start	timestamp
stop	timestamp
patient_id	varchar
encounter_id	varchar
code	integer
description	varchar
base_cost	integer
reasoncode	integer
reasondescription	varchar

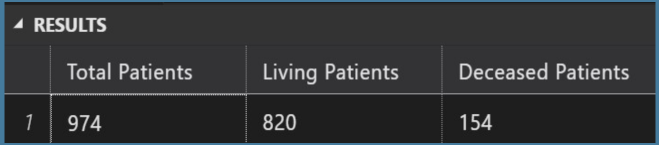
1. What is the total number of patients who are currently alive and the total number of patients who have deceased?

Query:



```sql
SELECT COUNT(*) AS 'Total Patients',
    SUM(CASE WHEN deathdate is null THEN 1 END) AS 'Living Patients',
    COUNT(deathdate) AS 'Deceased Patients'
FROM patients;
```

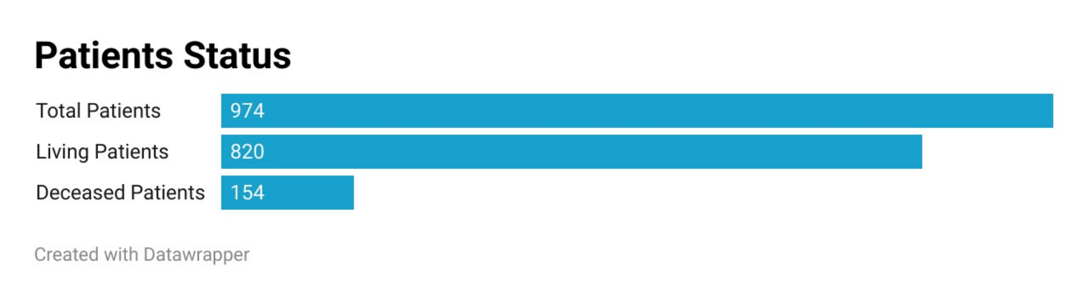
Output:



RESULTS

	Total Patients	Living Patients	Deceased Patients
1	974	820	154

Visual:

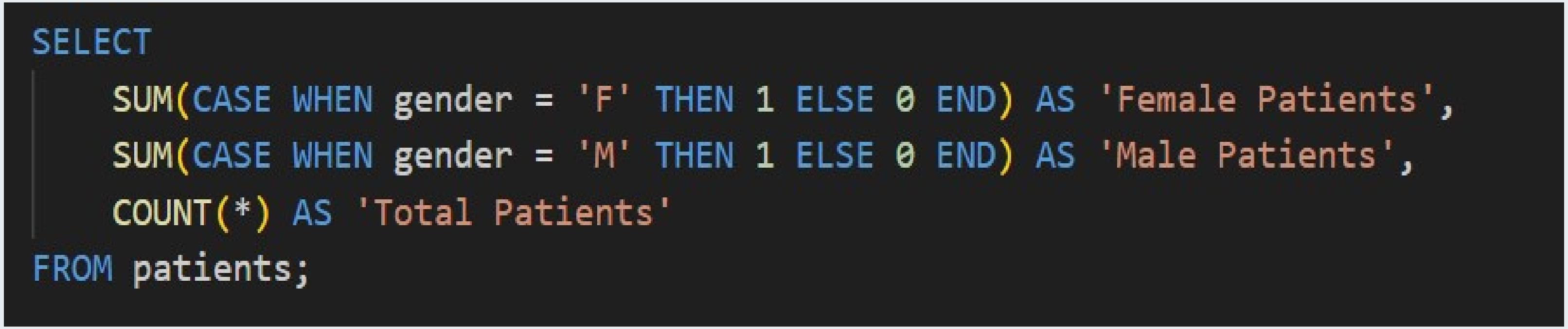


Insight:

Analyzing the count of alive and deceased patients helps hospital administrators assess patient outcomes and identify trends in mortality rates.

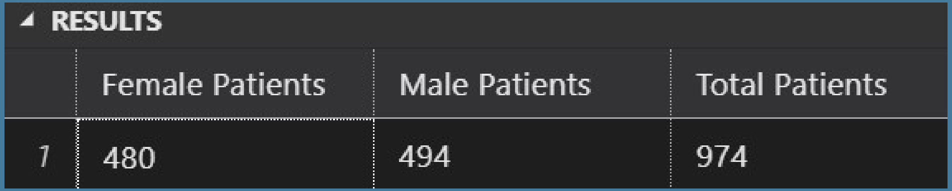
2. What is the distribution of patients by gender in the hospital?

Query:



```sql
SELECT
    SUM(CASE WHEN gender = 'F' THEN 1 ELSE 0 END) AS 'Female Patients',
    SUM(CASE WHEN gender = 'M' THEN 1 ELSE 0 END) AS 'Male Patients',
    COUNT(*) AS 'Total Patients'
FROM patients;
```

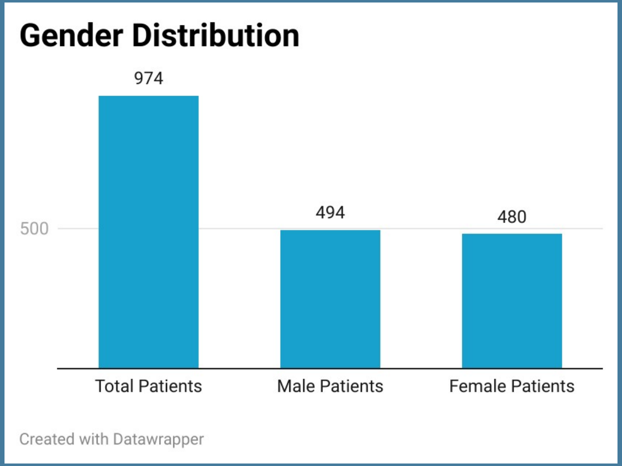
Output:



◢ RESULTS

	Female Patients	Male Patients	Total Patients
1	480	494	974

Visual:



Gender Distribution

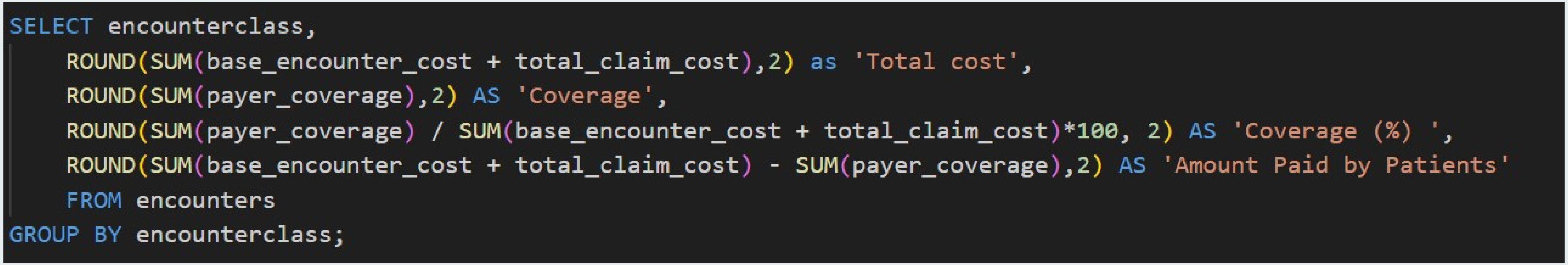
Created with Datawrapper

Insight:

Analyzing the count of alive and deceased patients helps hospital administrators assess patient outcomes and identify trends in mortality rates.

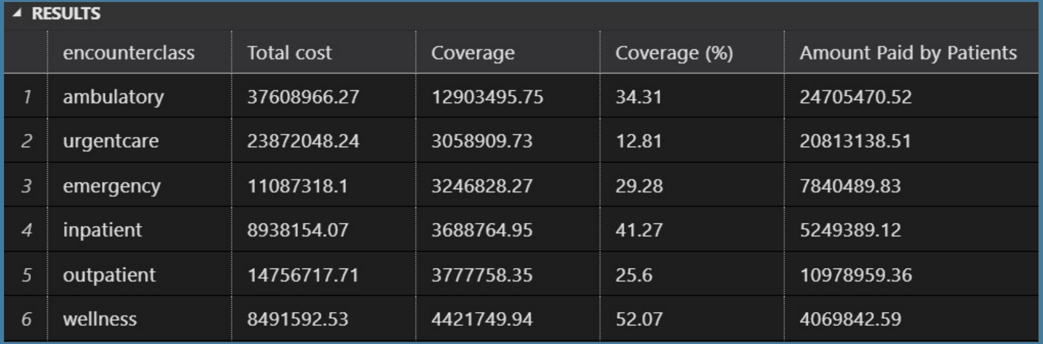
3. What is the total cost of care, insurance coverage, and patient-paid amounts for hospital services?

Query:



```
SELECT encounterclass,
    ROUND(SUM(base_encounter_cost + total_claim_cost),2) as 'Total cost',
    ROUND(SUM(payer_coverage),2) AS 'Coverage',
    ROUND(SUM(payer_coverage) / SUM(base_encounter_cost + total_claim_cost)*100, 2) AS 'Coverage (%) ',
    ROUND(SUM(base_encounter_cost + total_claim_cost) - SUM(payer_coverage),2) AS 'Amount Paid by Patients'
    FROM encounters
GROUP BY encounterclass;
```

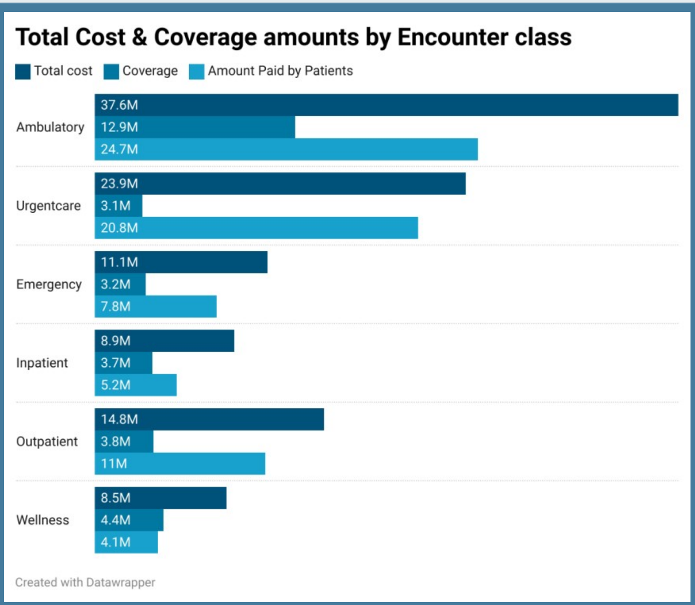
Output:



◢ RESULTS

	encounterclass	Total cost	Coverage	Coverage (%)	Amount Paid by Patients
1	ambulatory	37608966.27	12903495.75	34.31	24705470.52
2	urgentcare	23872048.24	3058909.73	12.81	20813138.51
3	emergency	11087318.1	3246828.27	29.28	7840489.83
4	inpatient	8938154.07	3688764.95	41.27	5249389.12
5	outpatient	14756717.71	3777758.35	25.6	10978959.36
6	wellness	8491592.53	4421749.94	52.07	4069842.59

Visual:



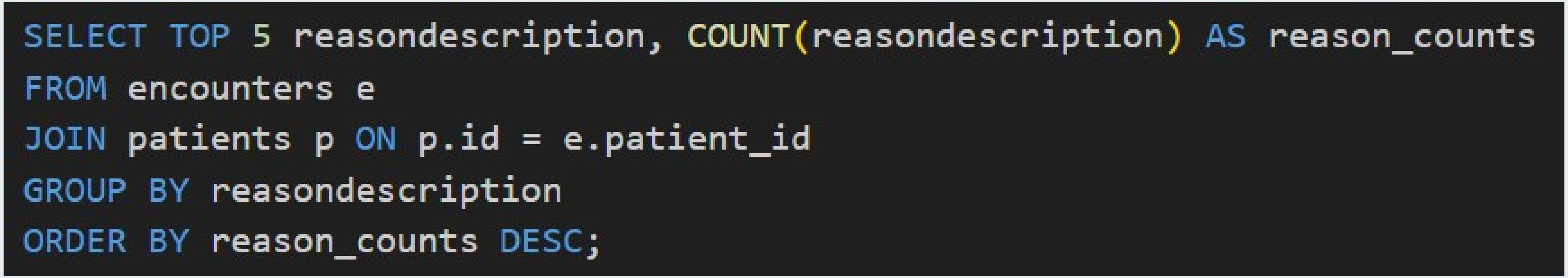
Total Cost & Coverage amounts by Encounter class

Created with Datawrapper

Insight: Understanding the financial breakdown of hospital services helps administrators identify areas for cost optimization, improve insurance reimbursement processes, and develop patient-centric financial assistance programs.

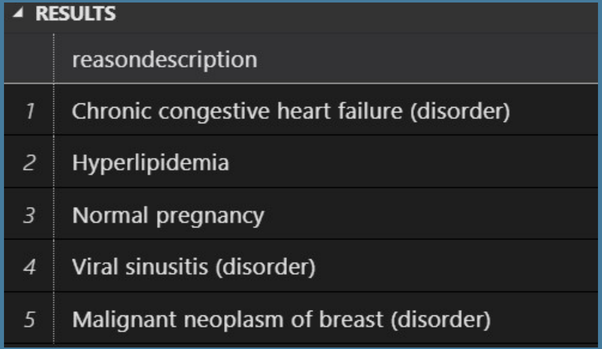
4. What are the top 5 encounter reasons for patient encounters in the hospital?

Query:



```sql
SELECT TOP 5 reasondescription, COUNT(reasondescription) AS reason_counts
FROM encounters e
JOIN patients p ON p.id = e.patient_id
GROUP BY reasondescription
ORDER BY reason_counts DESC;
```

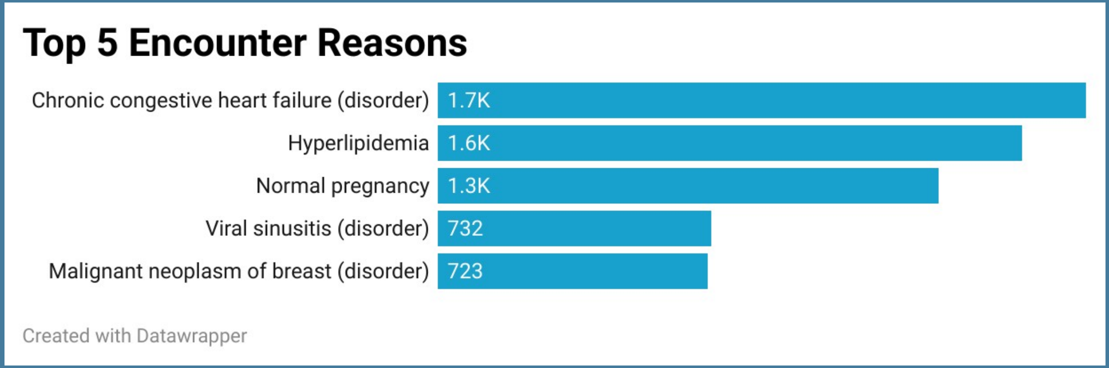
Output:



RESULTS

	reasondescription
1	Chronic congestive heart failure (disorder)
2	Hyperlipidemia
3	Normal pregnancy
4	Viral sinusitis (disorder)
5	Malignant neoplasm of breast (disorder)

Visual:



Top 5 Encounter Reasons

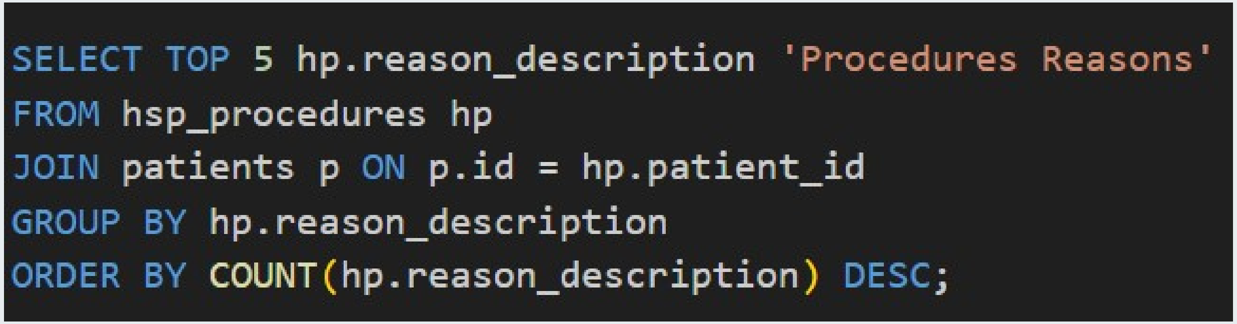
Created with Datawrapper

Insight:

Analyzing the top encounter reasons enables hospital administrators to identify trends and patterns in patient care, informing strategic decisions to improve patient outcomes and reduce readmissions.

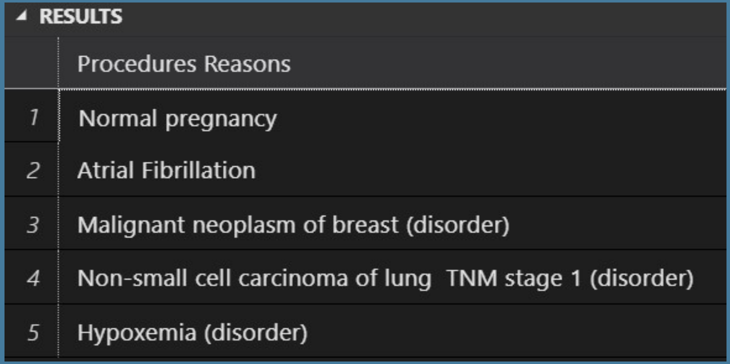
5. What are the top 5 procedures reasons for patient encounters in the hospital?

Query:



```
SELECT TOP 5 hp.reason_description 'Procedures Reasons'
FROM hsp_procedures hp
JOIN patients p ON p.id = hp.patient_id
GROUP BY hp.reason_description
ORDER BY COUNT(hp.reason_description) DESC;
```

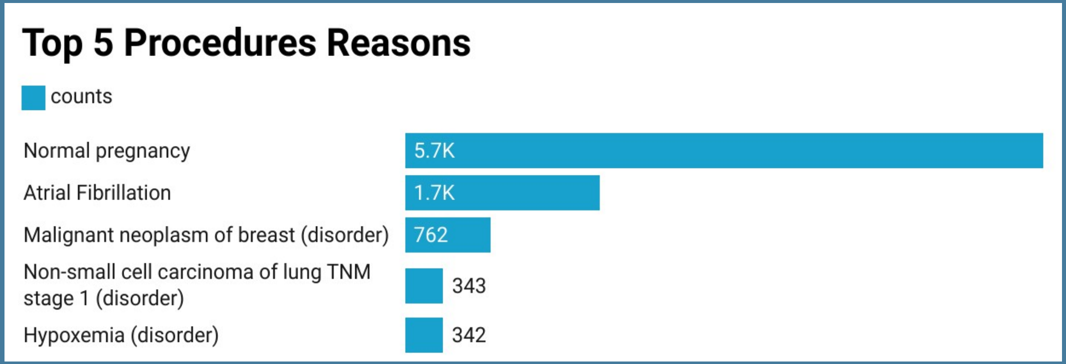
Output:



RESULTS

	Procedures Reasons
1	Normal pregnancy
2	Atrial Fibrillation
3	Malignant neoplasm of breast (disorder)
4	Non-small cell carcinoma of lung TNM stage 1 (disorder)
5	Hypoxemia (disorder)

Visual:

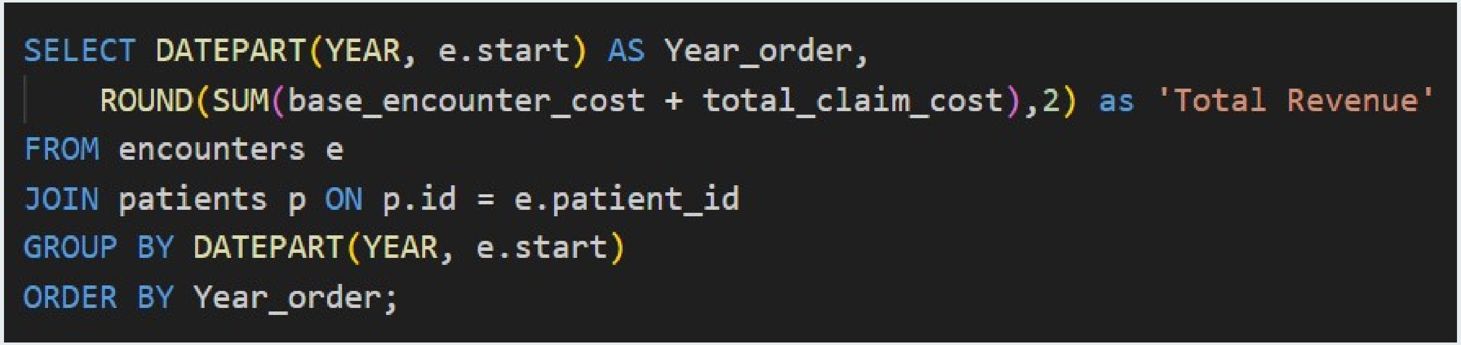


Insight:

Analyzing the top procedures reasons enables hospital administrators to identify trends and patterns in patient care, informing strategic decisions to improve patient outcomes and reduce readmissions.

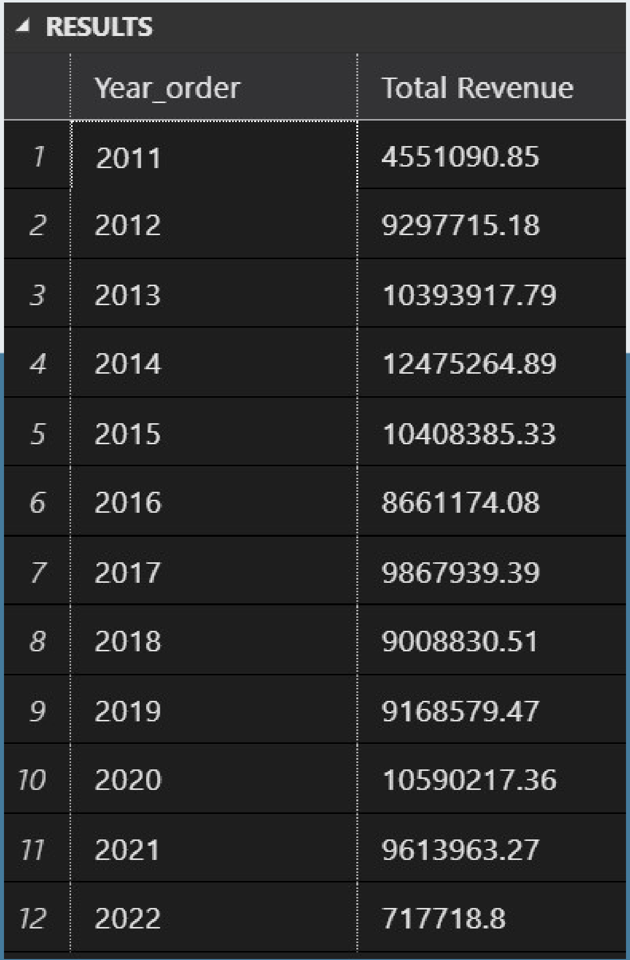
6. What is the total revenue trend for the hospital over the last 10 years?

Query:



```sql
SELECT DATEPART(YEAR, e.start) AS Year_order,
    ROUND(SUM(base_encounter_cost + total_claim_cost),2) as 'Total Revenue'
FROM encounters e
JOIN patients p ON p.id = e.patient_id
GROUP BY DATEPART(YEAR, e.start)
ORDER BY Year_order;
```

Output

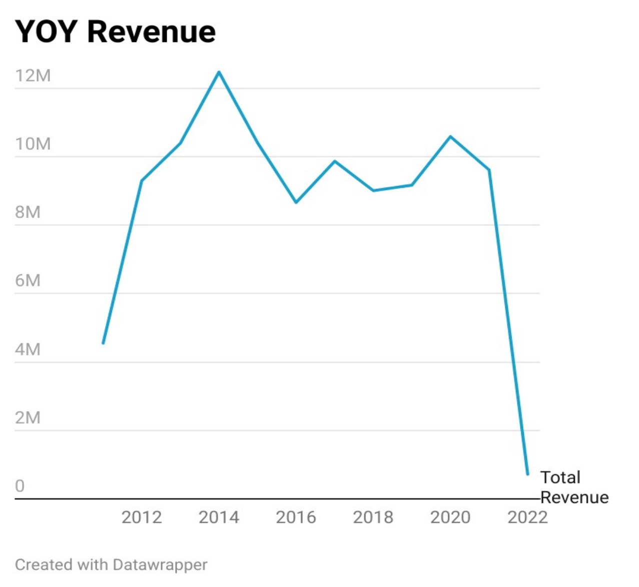


RESULTS

	Year_order	Total Revenue
1	2011	4551090.85
2	2012	9297715.18
3	2013	10393917.79
4	2014	12475264.89
5	2015	10408385.33
6	2016	8661174.08
7	2017	9867939.39
8	2018	9008830.51
9	2019	9168579.47
10	2020	10590217.36
11	2021	9613963.27
12	2022	717718.8

Insight:

Analyzing the 10-year revenue trend helps hospital administrators identify patterns of growth or decline.

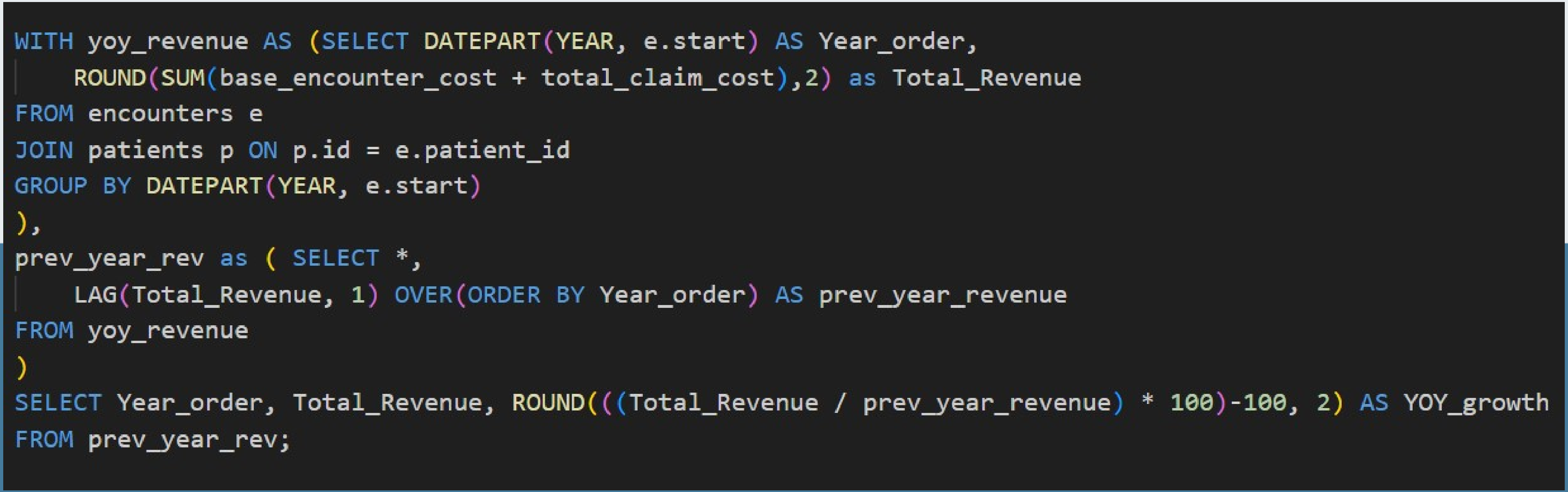


YOY Revenue

Created with Datawrapper

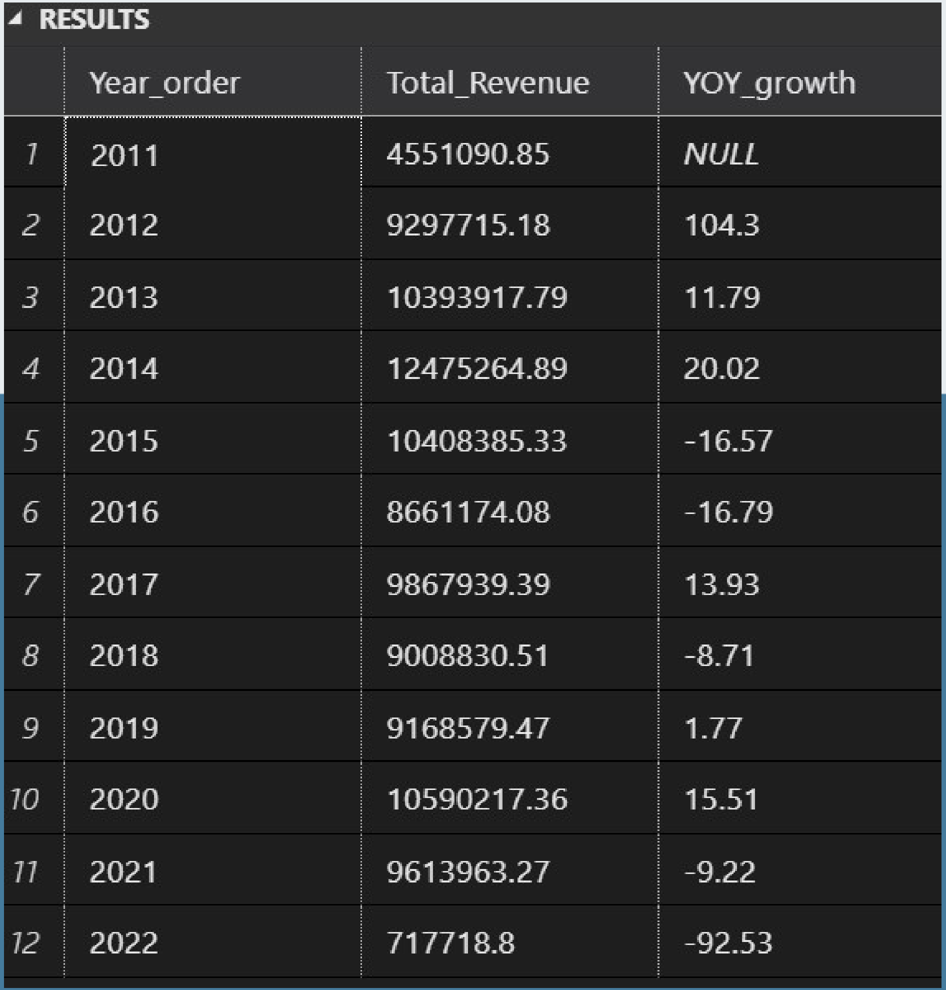
7. What is the year-over-year (YOY) revenue growth rate for the hospital over the last 10 years?

Query:



```sql
WITH yoy_revenue AS (SELECT DATEPART(YEAR, e.start) AS Year_order,
    ROUND(SUM(base_encounter_cost + total_claim_cost),2) as Total_Revenue
FROM encounters e
JOIN patients p ON p.id = e.patient_id
GROUP BY DATEPART(YEAR, e.start)
),
prev_year_rev as ( SELECT *,
    LAG(Total_Revenue, 1) OVER(ORDER BY Year_order) AS prev_year_revenue
FROM yoy_revenue
)
SELECT Year_order, Total_Revenue, ROUND(((Total_Revenue / prev_year_revenue) * 100)-100, 2) AS YOY_growth
FROM prev_year_rev;
```

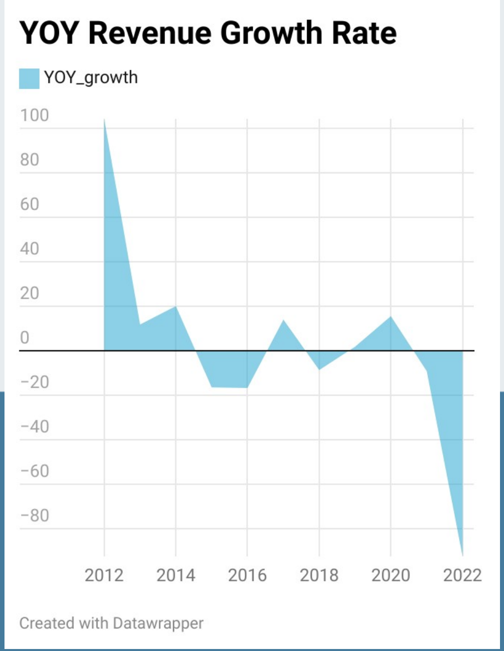
Output:



RESULTS

	Year_order	Total_Revenue	YOY_growth
1	2011	4551090.85	*NULL*
2	2012	9297715.18	104.3
3	2013	10393917.79	11.79
4	2014	12475264.89	20.02
5	2015	10408385.33	-16.57
6	2016	8661174.08	-16.79
7	2017	9867939.39	13.93
8	2018	9008830.51	-8.71
9	2019	9168579.47	1.77
10	2020	10590217.36	15.51
11	2021	9613963.27	-9.22
12	2022	717718.8	-92.53

Visual:



YOY Revenue Growth Rate

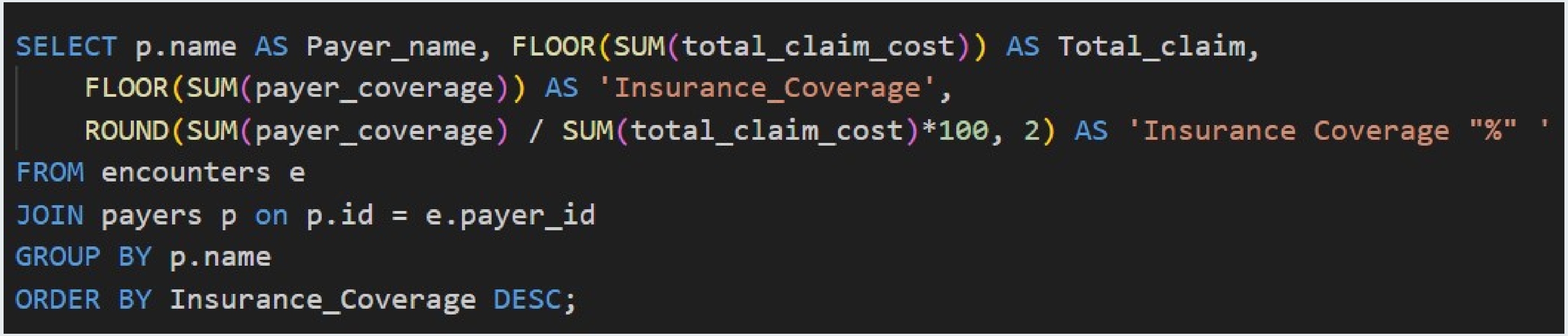
Created with Datawrapper

Insight:

The fluctuating YOY revenue growth rate reveals periods of expansion and contraction, enabling hospital administrators to identify factors influencing revenue volatility and make informed decisions to drive sustainable growth.

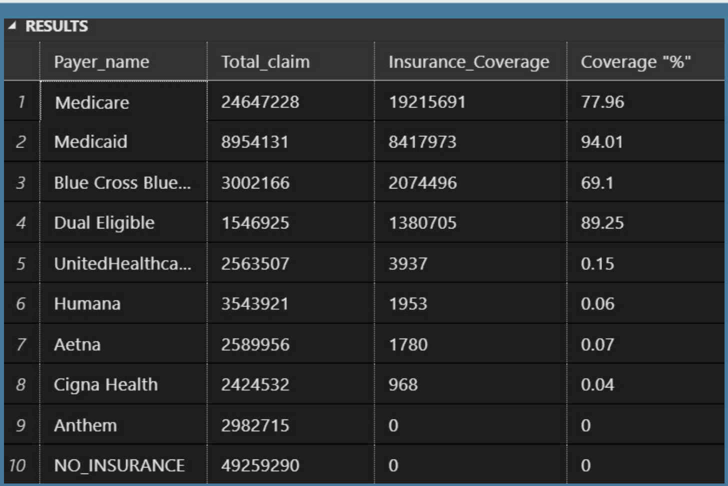
8. Which payer has the highest total claim amount and insurance coverage percentage?

Query:



```
SELECT p.name AS Payer_name, FLOOR(SUM(total_claim_cost)) AS Total_claim,
    FLOOR(SUM(payer_coverage)) AS 'Insurance_Coverage',
    ROUND(SUM(payer_coverage) / SUM(total_claim_cost)*100, 2) AS 'Insurance Coverage "%" '
FROM encounters e
JOIN payers p on p.id = e.payer_id
GROUP BY p.name
ORDER BY Insurance_Coverage DESC;
```

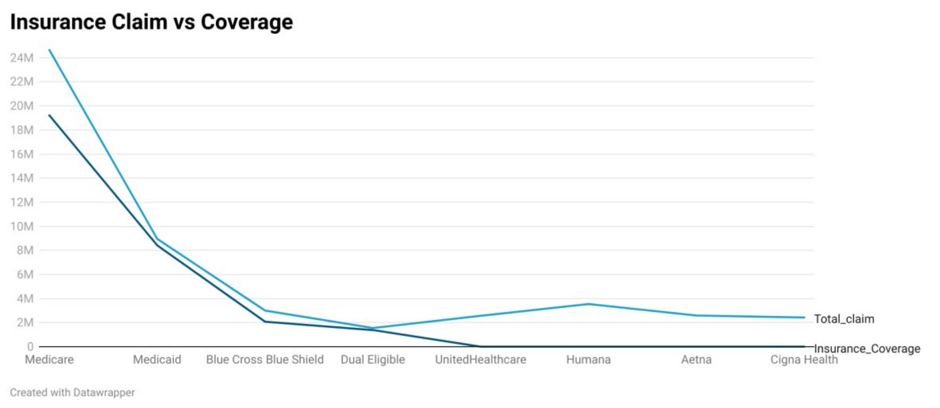
Output:



RESULTS

	Payer_name	Total_claim	Insurance_Coverage	Coverage "%"
1	Medicare	24647228	19215691	77.96
2	Medicaid	8954131	8417973	94.01
3	Blue Cross Blue...	3002166	2074496	69.1
4	Dual Eligible	1546925	1380705	89.25
5	UnitedHealthca...	2563507	3937	0.15
6	Humana	3543921	1953	0.06
7	Aetna	2589956	1780	0.07
8	Cigna Health	2424532	968	0.04
9	Anthem	2982715	0	0
10	NO_INSURANCE	49259290	0	0

Visual:



Insurance Claim vs Coverage

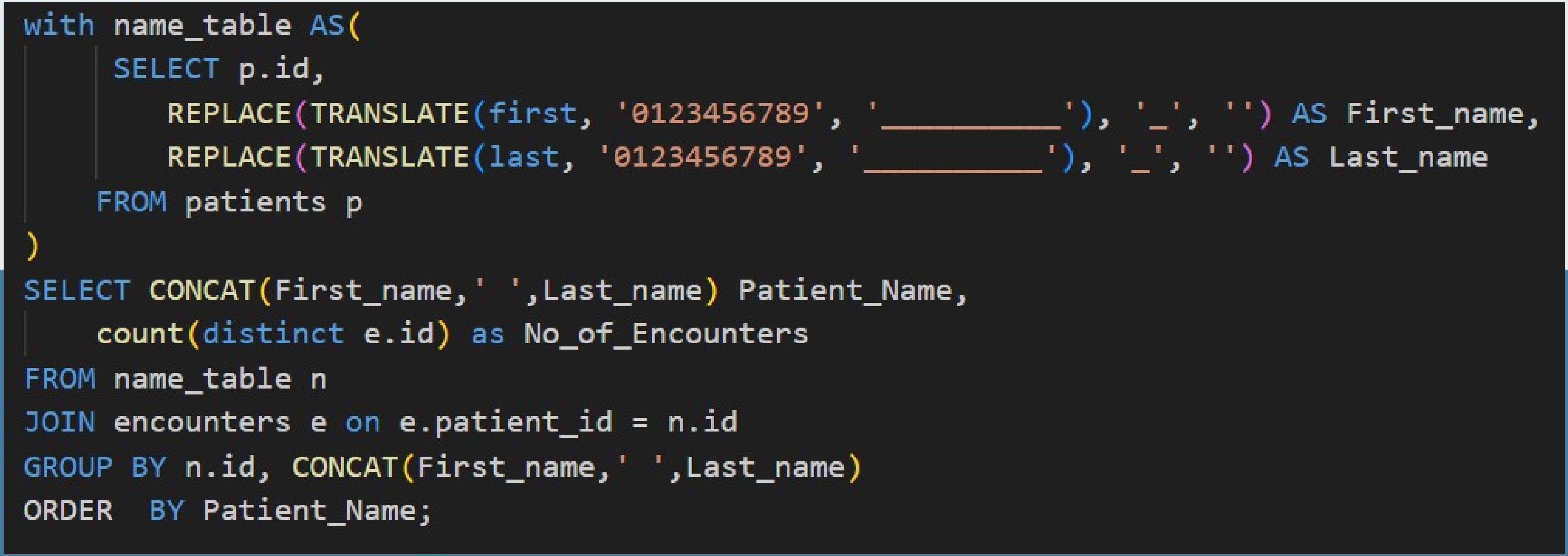
Created with Datawrapper

Insight:

Analyzing the gap between total claims and covered amounts helps hospital administrators identify trends in insurance reimbursement, optimize billing processes, and negotiate better coverage terms with insurance providers.

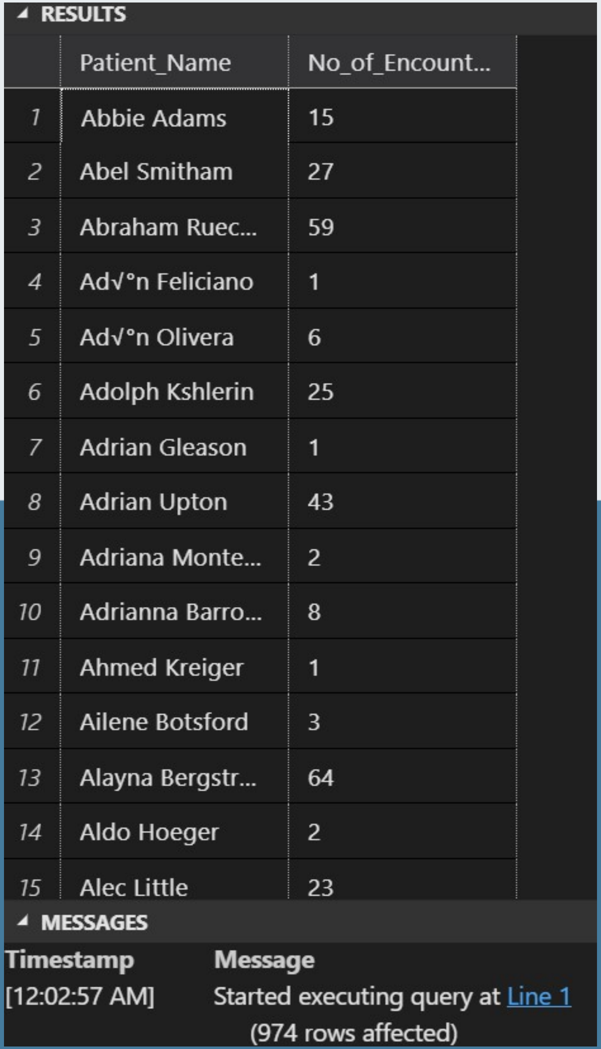
9. What is the full name of each patient and how many encounters have they had at the hospital?

Query:



```sql
with name_table AS(
     SELECT p.id,
        REPLACE(TRANSLATE(first, '0123456789', '__________'), '_', '') AS First_name,
        REPLACE(TRANSLATE(last, '0123456789', '__________'), '_', '') AS Last_name
    FROM patients p
)
SELECT CONCAT(First_name,' ',Last_name) Patient_Name,
    count(distinct e.id) as No_of_Encounters
FROM name_table n
JOIN encounters e on e.patient_id = n.id
GROUP BY n.id, CONCAT(First_name,' ',Last_name)
ORDER  BY Patient_Name;
```

Query:



RESULTS

	Patient_Name	No_of_Encount...
1	Abbie Adams	15
2	Abel Smitham	27
3	Abraham Ruec...	59
4	Ad√°n Feliciano	1
5	Ad√°n Olivera	6
6	Adolph Kshlerin	25
7	Adrian Gleason	1
8	Adrian Upton	43
9	Adriana Monte...	2
10	Adrianna Barro...	8
11	Ahmed Kreiger	1
12	Ailene Botsford	3
13	Alayna Bergstr...	64
14	Aldo Hoeger	2
15	Alec Little	23

MESSAGES

```
Timestamp      Message
[12:02:57 AM]  Started executing query at Line 1
                 (974 rows affected)
```

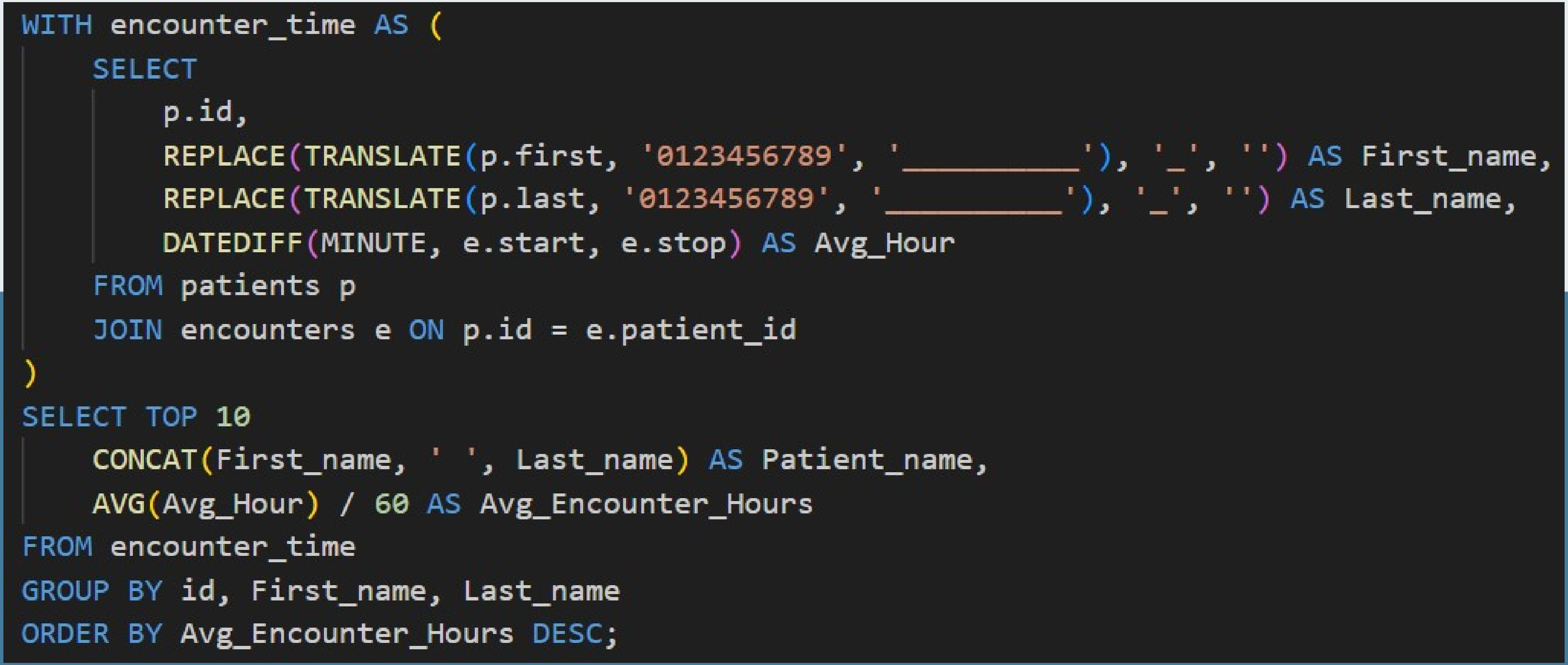
Output:

Insight:

Accurately tracking patient encounters enables hospital administrators to identify high-utilization patterns, optimize resource allocation, and inform quality improvement initiatives to enhance patient care and satisfaction.

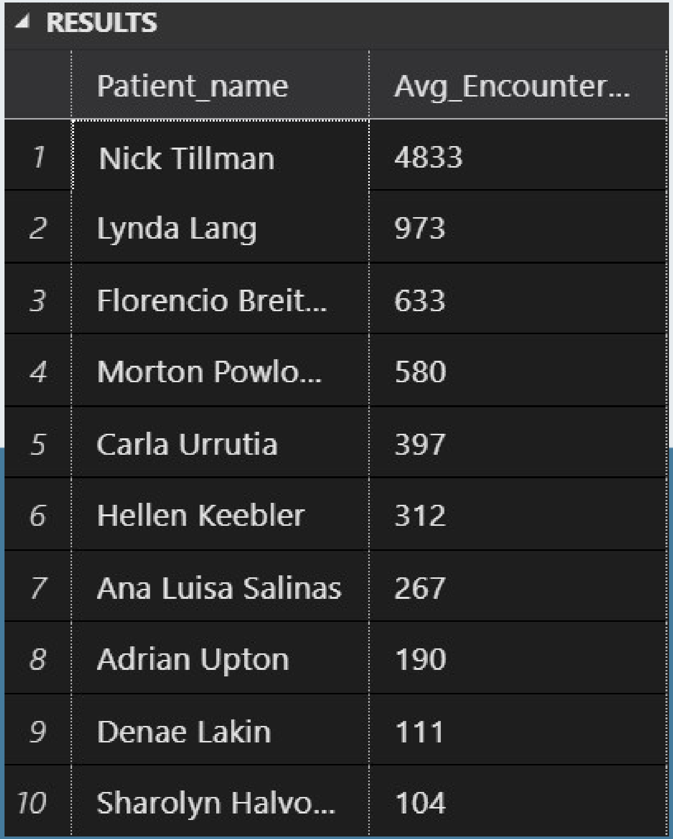
10. Which patients have spent an average of more than 100 hours of encounters in the hospital?

Query:



```sql
WITH encounter_time AS (
    SELECT
        p.id,
        REPLACE(TRANSLATE(p.first, '0123456789', '__________'), '_', '') AS First_name,
        REPLACE(TRANSLATE(p.last, '0123456789', '__________'), '_', '') AS Last_name,
        DATEDIFF(MINUTE, e.start, e.stop) AS Avg_Hour
    FROM patients p
    JOIN encounters e ON p.id = e.patient_id
)
SELECT TOP 10
    CONCAT(First_name, ' ', Last_name) AS Patient_name,
    AVG(Avg_Hour) / 60 AS Avg_Encounter_Hours
FROM encounter_time
GROUP BY id, First_name, Last_name
ORDER BY Avg_Encounter_Hours DESC;
```

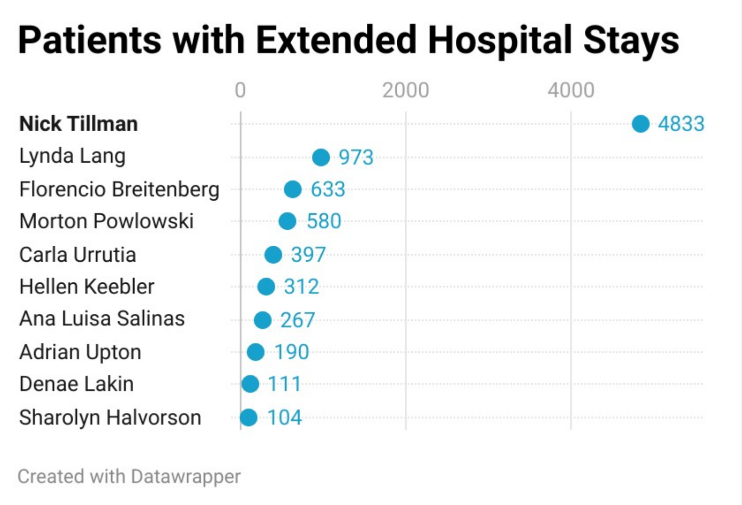
Output:



RESULTS

	Patient_name	Avg_Encounter...
1	Nick Tillman	4833
2	Lynda Lang	973
3	Florencio Breit...	633
4	Morton Powlo...	580
5	Carla Urrutia	397
6	Hellen Keebler	312
7	Ana Luisa Salinas	267
8	Adrian Upton	190
9	Denae Lakin	111
10	Sharolyn Halvo...	104

Visual:



Patients with Extended Hospital Stays

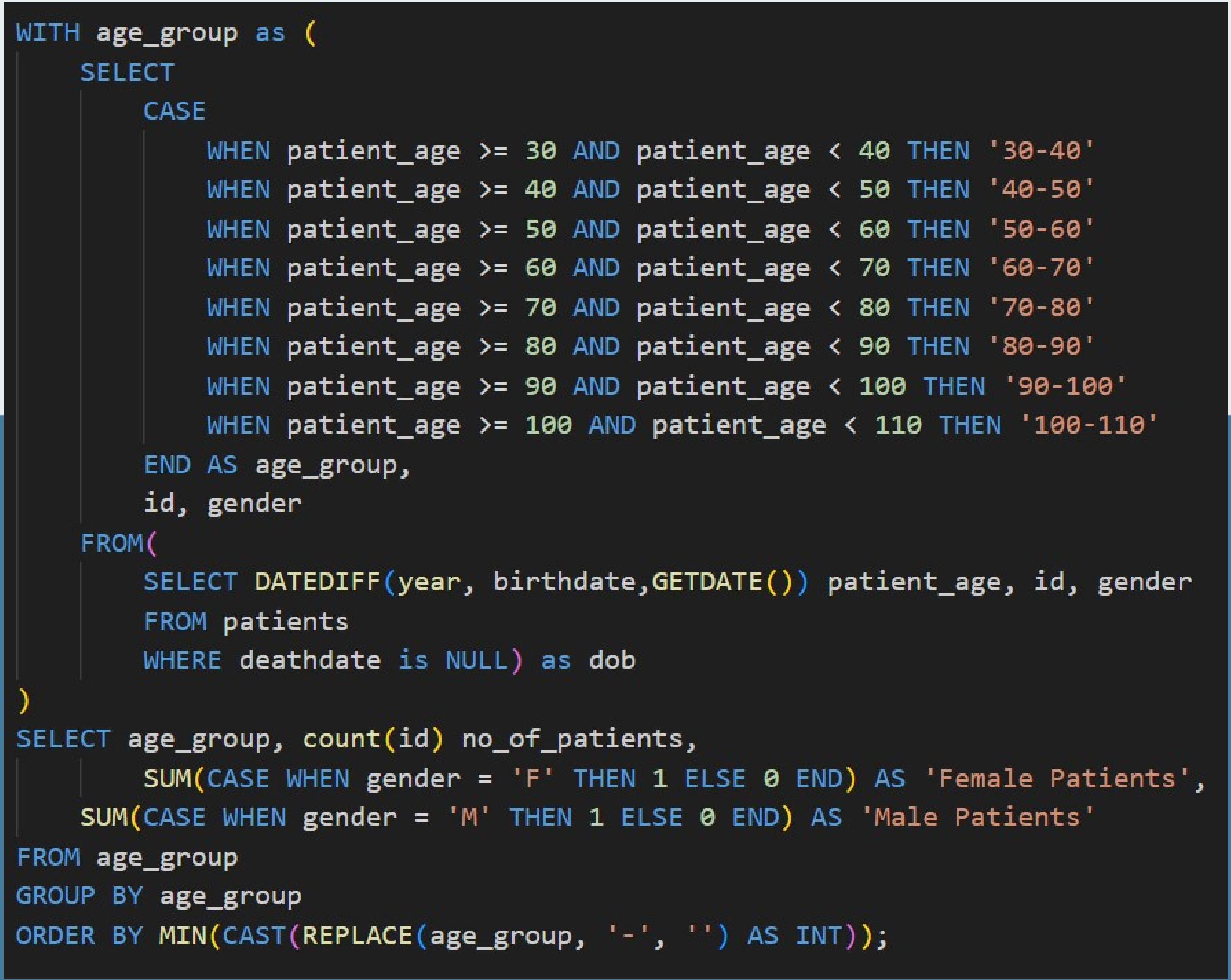
Created with Datawrapper

Insight:

Identifying patients with prolonged encounters enables hospital administrators to optimize resource allocation, improve patient flow, and enhance care quality, ultimately reducing healthcare costs and informing data-driven decisions.

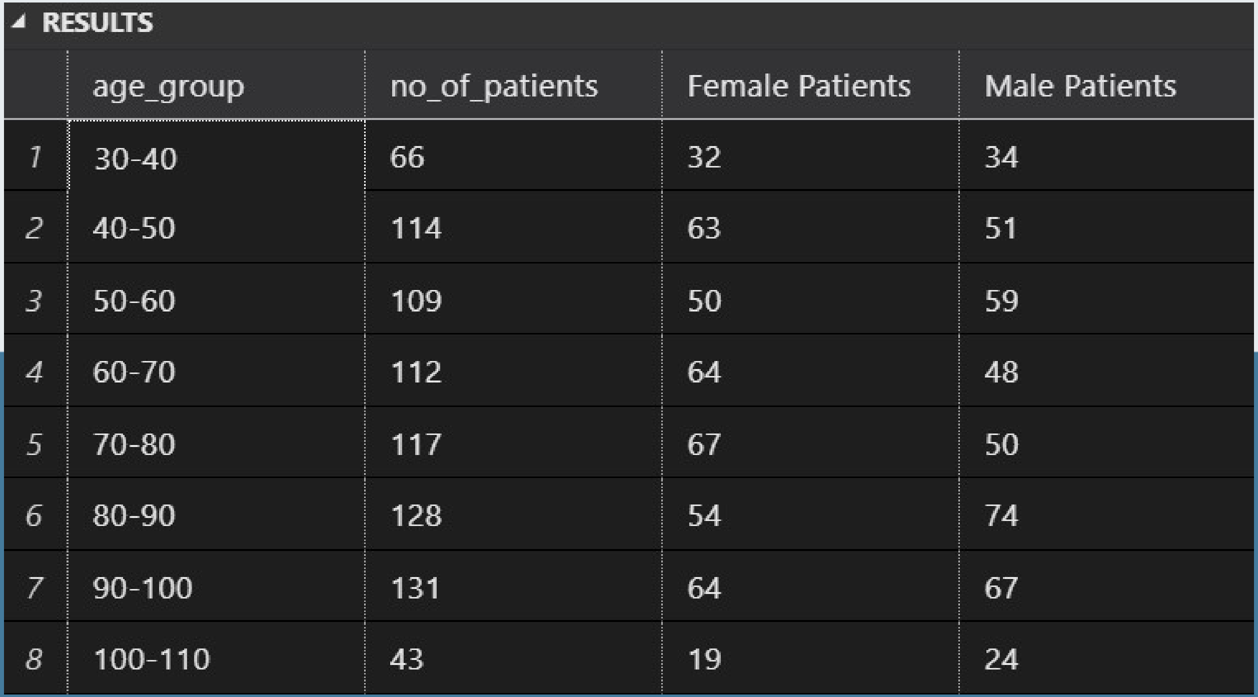
11. What is the distribution of patients across different groups (e.g., age, gender in the hospital?

Query:



```
WITH age_group as (
    SELECT
        CASE
            WHEN patient_age >= 30 AND patient_age < 40 THEN '30-40'
            WHEN patient_age >= 40 AND patient_age < 50 THEN '40-50'
            WHEN patient_age >= 50 AND patient_age < 60 THEN '50-60'
            WHEN patient_age >= 60 AND patient_age < 70 THEN '60-70'
            WHEN patient_age >= 70 AND patient_age < 80 THEN '70-80'
            WHEN patient_age >= 80 AND patient_age < 90 THEN '80-90'
            WHEN patient_age >= 90 AND patient_age < 100 THEN '90-100'
            WHEN patient_age >= 100 AND patient_age < 110 THEN '100-110'
        END AS age_group,
        id, gender
    FROM(
        SELECT DATEDIFF(year, birthdate,GETDATE()) patient_age, id, gender
        FROM patients
        WHERE deathdate is NULL) as dob
)
SELECT age_group, count(id) no_of_patients,
        SUM(CASE WHEN gender = 'F' THEN 1 ELSE 0 END) AS 'Female Patients',
    SUM(CASE WHEN gender = 'M' THEN 1 ELSE 0 END) AS 'Male Patients'
FROM age_group
GROUP BY age_group
ORDER BY MIN(CAST(REPLACE(age_group, '-', '') AS INT));
```

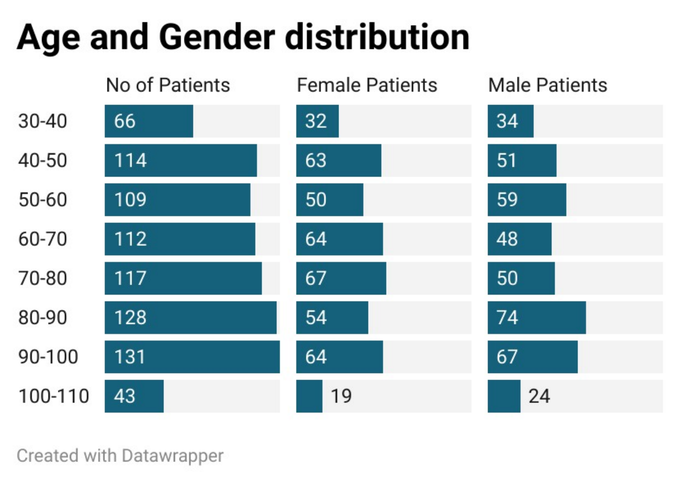
Output:



RESULTS

	age_group	no_of_patients	Female Patients	Male Patients
1	30-40	66	32	34
2	40-50	114	63	51
3	50-60	109	50	59
4	60-70	112	64	48
5	70-80	117	67	50
6	80-90	128	54	74
7	90-100	131	64	67
8	100-110	43	19	24

Visual:

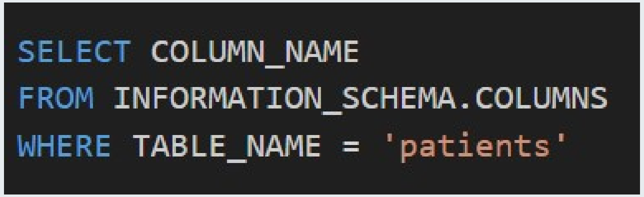


Insight:

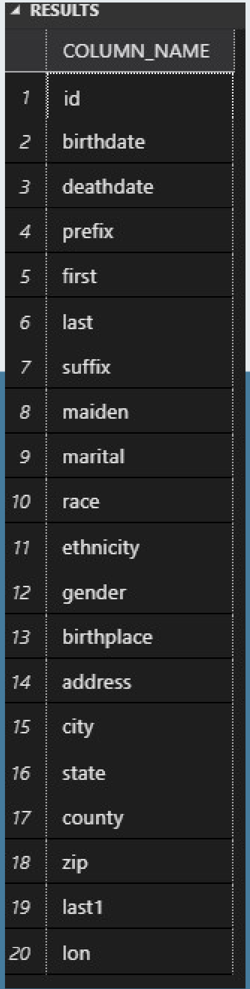
Patient distribution insights help identify areas for quality improvement, enabling focused initiatives to enhance patient care and satisfaction.

12. Getting Column names before altering table

Query:



```sql
SELECT COLUMN_NAME
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'patients'
```

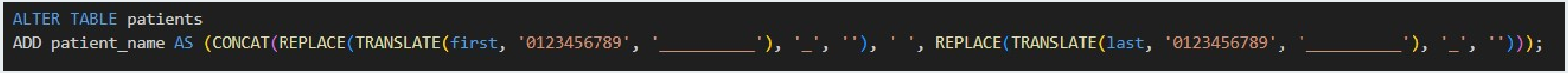


RESULTS

	COLUMN_NAME
1	id
2	birthdate
3	deathdate
4	prefix
5	first
6	last
7	suffix
8	maiden
9	marital
10	race
11	ethnicity
12	gender
13	birthplace
14	address
15	city
16	state
17	county
18	zip
19	last1
20	lon

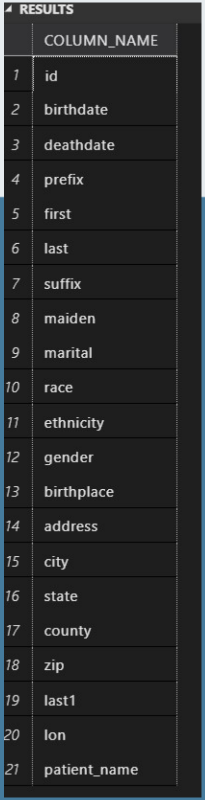
13. Altering table patients to get their full name from the first and last name columns

Query:



```
ALTER TABLE patients
ADD patient_name AS (CONCAT(REPLACE(TRANSLATE(first, '0123456789', '__________'), '_', ''), ' ', REPLACE(TRANSLATE(last, '0123456789', '__________'), '_', '')));
```

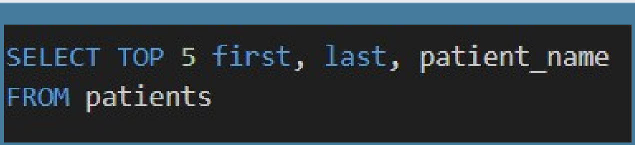
Before altering it, there were 20 columns; now we have 21 columns.



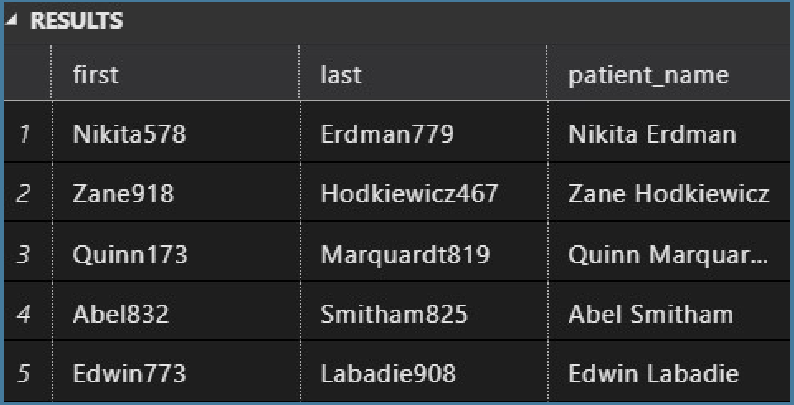
RESULTS

	COLUMN_NAME
1	id
2	birthdate
3	deathdate
4	prefix
5	first
6	last
7	suffix
8	maiden
9	marital
10	race
11	ethnicity
12	gender
13	birthplace
14	address
15	city
16	state
17	county
18	zip
19	last1
20	lon
21	patient_name

First, Last, Full name from table patients



```
SELECT TOP 5 first, last, patient_name
FROM patients
```

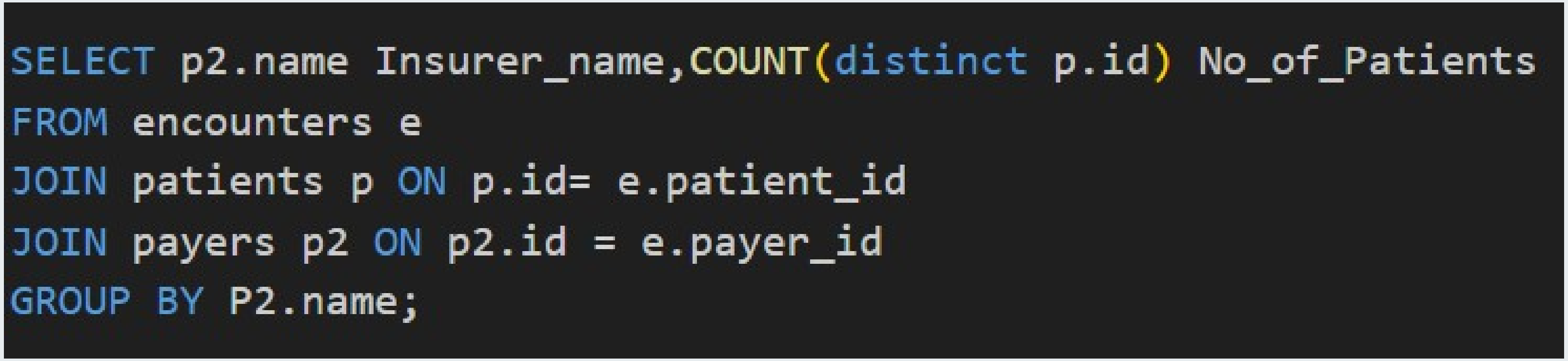


RESULTS

	first	last	patient_name
1	Nikita578	Erdman779	Nikita Erdman
2	Zane918	Hodkiewicz467	Zane Hodkiewicz
3	Quinn173	Marquardt819	Quinn Marquar...
4	Abel832	Smitham825	Abel Smitham
5	Edwin773	Labadie908	Edwin Labadie

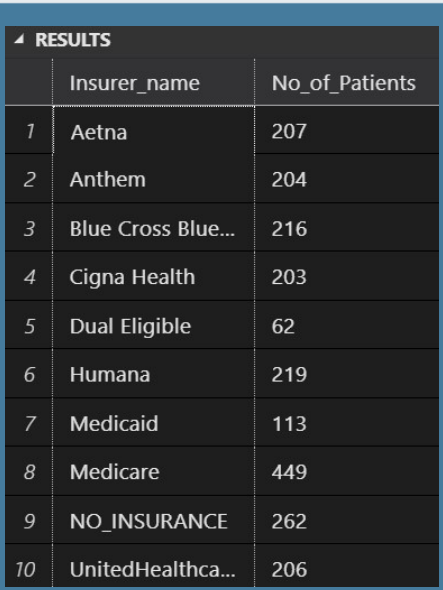
14. What is the distribution of patients across different payers

Query:



```
SELECT p2.name Insurer_name,COUNT(distinct p.id) No_of_Patients
FROM encounters e
JOIN patients p ON p.id= e.patient_id
JOIN payers p2 ON p2.id = e.payer_id
GROUP BY P2.name;
```

Output:



RESULTS

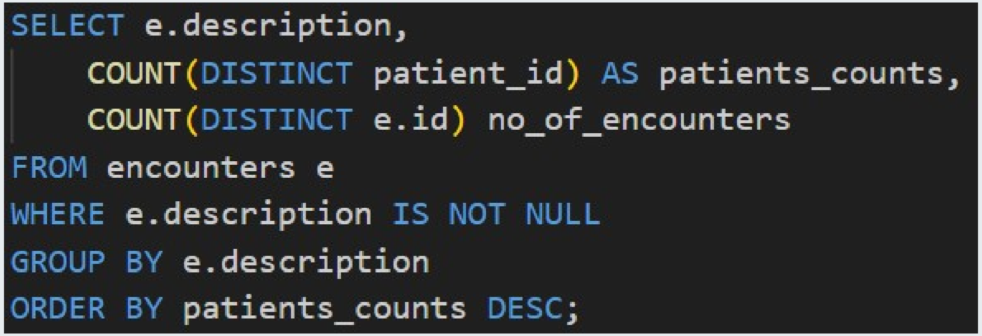
	Insurer_name	No_of_Patients
1	Aetna	207
2	Anthem	204
3	Blue Cross Blue...	216
4	Cigna Health	203
5	Dual Eligible	62
6	Humana	219
7	Medicaid	113
8	Medicare	449
9	NO_INSURANCE	262
10	UnitedHealthca...	206

Insight:

Analyzing the distribution of patients across payers reveals opportunities for hospitals to refine market strategies, improve patient retention, and negotiate more favorable reimbursement rates with dominant payers.

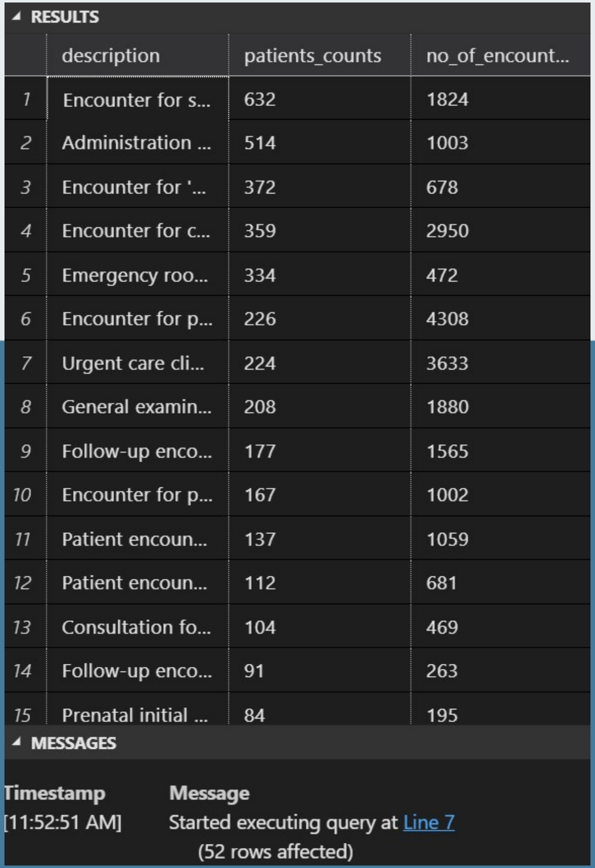
15. What are the top patient encounter descriptions, associated patient counts, and encounter frequencies

Query:



```sql
SELECT e.description,
    COUNT(DISTINCT patient_id) AS patients_counts,
    COUNT(DISTINCT e.id) no_of_encounters
FROM encounters e
WHERE e.description IS NOT NULL
GROUP BY e.description
ORDER BY patients_counts DESC;
```

Output:



RESULTS

	description	patients_counts	no_of_encount...
1	Encounter for s...	632	1824
2	Administration ...	514	1003
3	Encounter for '...	372	678
4	Encounter for c...	359	2950
5	Emergency roo...	334	472
6	Encounter for p...	226	4308
7	Urgent care cli...	224	3633
8	General examin...	208	1880
9	Follow-up enco...	177	1565
10	Encounter for p...	167	1002
11	Patient encoun...	137	1059
12	Patient encoun...	112	681
13	Consultation fo...	104	469
14	Follow-up enco...	91	263
15	Prenatal initial ...	84	195

MESSAGES

Timestamp	Message
[11:52:51 AM]	Started executing query at Line 7 (52 rows affected)

Note: This analysis considers a 10-year dataset, where the same patients may have been readmitted multiple times for the same problem, resulting in a higher patient count.

Insight:

Analyzing patient encounter descriptions, patient counts, and encounter frequencies reveals patterns of care utilization, readmission rates, and potential areas for quality improvement, enabling hospitals to optimize care pathways, reduce readmissions, and enhance patient outcomes.

MGH
1811

THANK YOU

Thanks for checking out this project.

Connect with me on *LinkedIn*

For more details of this project, *Click here* to navigate to my GitHub portfolio