In [2]:
```
!pip install pymongo
```

```
Collecting pymongo
  Downloading pymongo-4.13.0-cp39-cp39-win_amd64.whl (747 kB)
Collecting dnspython<3.0.0,>=1.16.0
  Downloading dnspython-2.7.0-py3-none-any.whl (313 kB)
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.7.0 pymongo-4.13.0
```

In [5]:
```
import pandas as pd
from pymongo import MongoClient
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from datetime import datetime

print("Connexion à MongoDB...")
client = MongoClient("mongodb://admin:admin@localhost:27017/")
db = client.healthcare_db
collection = db.patients

count = collection.count_documents({})
print(f"Nombre de documents dans la collection : {count}")
```

```
Connexion à MongoDB...
Nombre de documents dans la collection : 55500
```

In [4]:
```
print("Question 1: How many patients are in the collection?")
total_patients = collection.count_documents({})
print(f"Total number of patients: {total_patients}")
```

```
Question 1: How many patients are in the collection?
Total number of patients: 55500
```

In [9]:
```
print("Question 2: List all patients admitted after January 1, 2023")

patients_after_2023 = list(collection.find({
    "Date of Admission": {"$gt": datetime(2023, 1, 1)}
}))

print(f"Number of patients admitted after January 1, 2023: {len(patients_after_2023)}")
```

```
Question 2: List all patients admitted after January 1, 2023
Number of patients admitted after January 1, 2023: 14848
Number of patients admitted after January 1, 2023: 14848
```

In [10]:
```python
print("Question 3:")

patients_over_50 = collection.count_documents({
    "Age": {"$gt": 50}
})
print(f"Patients older than 50: {patients_over_50}")

patients_thomas = collection.count_documents({
    "Name": {"$regex": "^Thomas "}
})
print(f"Patients with first name 'Thomas': {patients_thomas}")

print("\nPatients per each distinct Medical Condition:")
pipeline = [
    {"$group": {"_id": "$Medical Condition", "count": {"$sum": 1}}},
    {"$sort": {"count": -1}}
]

conditions = list(collection.aggregate(pipeline))
for condition in conditions:
    print(f"   {condition['_id']}: {condition['count']} patients")
```

```
Question 3:
Patients older than 50: 28667
Patients with first name 'Thomas': 397

Patients per each distinct Medical Condition:
   Arthritis: 9308 patients
   Diabetes: 9304 patients
   Hypertension: 9245 patients
   Obesity: 9231 patients
   Cancer: 9227 patients
   Asthma: 9185 patients
```

In [13]:
```python
print("Question 4: What is the frequency of usage for each Medication?")

medication_pipeline = [
    {"$group": {"_id": "$Medication", "frequency": {"$sum": 1}}},
    {"$sort": {"frequency": -1}}
]
```

```
medications = list(collection.aggregate(medication_pipeline))

print(f"Total distinct medications: {len(medications)}")
print("\nMedication frequency 10:")
for i, med in enumerate(medications[:10]):
    print(f"{i+1:2}. {med['_id']}: {med['frequency']} patients")
```

```
Question 4: What is the frequency of usage for each Medication?
Total distinct medications: 5

Medication frequency 10:
 1. Lipitor: 11140 patients
 2. Ibuprofen: 11127 patients
 3. Aspirin: 11094 patients
 4. Paracetamol: 11071 patients
 5. Penicillin: 11068 patients
```

In [14]:

```
print("Question 5: Retrieve all patients currently taking 'Lipitor'")

lipitor_patients = list(collection.find({
    "Medication": "Lipitor"
}))

print(f"Number of patients taking Lipitor: {len(lipitor_patients)}")

print("\nFirst 5 patients taking Lipitor:")
for i, patient in enumerate(lipitor_patients[:5]):
    print(f"{i+1}. {patient['Name']} (Age: {patient['Age']}) - {patient['Medical Condition']}")
```

```
Question 5: Retrieve all patients currently taking 'Lipitor'
Number of patients taking Lipitor: 11140

First 5 patients taking Lipitor:
1. Aaron Martinez (Age: 38) - Hypertension
2. Robert Bauer (Age: 68) - Asthma
3. Christopher Bright (Age: 48) - Asthma
4. Kathryn Stewart (Age: 58) - Arthritis
5. Dr. Eileen Thompson (Age: 59) - Asthma
```