In [3]:
```
from dotenv import load_dotenv
import os

# Load environment variables from .env file
load_dotenv()

NEO4J_USER = os.getenv("NEO4J_USER")
NEO4J_PASSWORD = os.getenv("NEO4J_PASS")
openai_api_key = os.getenv("openai_api_key")
```

In [ ]:
```
from google.cloud import bigquery
client = bigquery.Client()


admission_query = """
WITH eligible_patients AS (
  SELECT subject_id
  FROM `physionet-data.mimiciii_clinical.admissions`
  GROUP BY subject_id
  HAVING COUNT(DISTINCT hadm_id) BETWEEN 1 AND 3
),
sampled_patients AS (
  SELECT subject_id
  FROM eligible_patients
  ORDER BY RAND()
  LIMIT 1000
)
SELECT a.*
FROM `physionet-data.mimiciii_clinical.admissions` a
JOIN sampled_patients s ON a.subject_id = s.subject_id
WHERE a.hadm_id IS NOT NULL
"""

sampled_patients_admissions_df = client.query(admission_query).to_dataframe(

# Formating the date columns
sampled_patients_admissions_df['ADMITTIME'] = pd.to_datetime(sampled_patient
sampled_patients_admissions_df['DISCHTIME'] = pd.to_datetime(sampled_patient

# Flag readmissions within 30 days
sampled_patients_admissions_df = sampled_patients_admissions_df.sort_values(
sampled_patients_admissions_df["NEXT_ADMITTIME"] = sampled_patients_admissio
sampled_patients_admissions_df["DAYS_TO_NEXT"] = (sampled_patients_admission
sampled_patients_admissions_df["READMIT_30D"] = sampled_patients_admissions_


sampled_hadm_ids = sampled_patients_admissions_df["HADM_ID"].unique().tolist
sampled_patient_ids = sampled_patients_admissions_df['SUBJECT_ID'].unique().

# Convert list of Subject and HADM IDs into a SQL IN clause
id_list_str = ", ".join(str(x) for x in sampled_patient_ids)
id_filter = f"({id_list_str})"
```

```
hadm_list_str = ", ".join(str(x) for x in sampled_hadm_ids)
hadm_filter = f"({hadm_list_str})"
```

In [106…

```
patient_query = """
SELECT *
FROM `physionet-data.mimiciii_clinical.patients` p
WHERE p.subject_id IN {id_filter}
"""
# Download the sampled patients data to CSV
sampled_patients_df = client.query(patient_query.format(id_filter=id_filter)
sampled_patients_df.to_csv("./neo4j-community-2025.03.0/import/patients.csv"

# Add AGE column to admissions dataframe
sampled_patients_admissions_df = sampled_patients_admissions_df.merge(sample

# Compute age at admission; Account for MIMIC III obfuscation of ages over 8
sampled_patients_admissions_df["AGE"] = (sampled_patients_admissions_df["ADM
sampled_patients_admissions_df["AGE"] = sampled_patients_admissions_df["AGE"

# Drop DOB column after use
sampled_patients_admissions_df = sampled_patients_admissions_df.drop(columns

# Download the sampled admissions data to CSV
sampled_patients_admissions_df.to_csv("./neo4j-community-2025.03.0/import/ad
```

```
/Users/abemankavil/Desktop/MS in AI/AI395T-AIinHealthcare/HighRisk_Project/H
ighRiskEnv/lib/python3.9/site-packages/google/cloud/bigquery/table.py:1933:
UserWarning: BigQuery Storage module not found, fetch data with the REST end
point instead.
  warnings.warn(
```

In [107…

```
diag_query = """
SELECT d.subject_id, d.hadm_id, d.icd9_code, icd.short_title AS diagnosis
FROM `physionet-data.mimiciii_clinical.diagnoses_icd` d
JOIN `physionet-data.mimiciii_clinical.d_icd_diagnoses` icd
  ON d.icd9_code = icd.icd9_code
WHERE d.hadm_id IN {hadm_filter}
"""

diag_df = client.query(diag_query.format(hadm_filter=hadm_filter)).to_datafr

diag_df.to_csv("./neo4j-community-2025.03.0/import/diagnoses.csv", index=Fal
```

```
/Users/abemankavil/Desktop/MS in AI/AI395T-AIinHealthcare/HighRisk_Project/H
ighRiskEnv/lib/python3.9/site-packages/google/cloud/bigquery/table.py:1933:
UserWarning: BigQuery Storage module not found, fetch data with the REST end
point instead.
  warnings.warn(
```

In [108…

```
drug_query = """
SELECT subject_id, hadm_id, drug, route, startdate, enddate, dose_val_rx, do
FROM `physionet-data.mimiciii_clinical.prescriptions`
WHERE hadm_id IN {hadm_filter}
AND drug IS NOT NULL
"""
```

```
drug_df = client.query(drug_query.format(hadm_filter=hadm_filter)).to_datafr
drug_df.to_csv("./neo4j-community-2025.03.0/import/prescriptions.csv", index
```

```
/Users/abemankavil/Desktop/MS in AI/AI395T-AIinHealthcare/HighRisk_Project/H
ighRiskEnv/lib/python3.9/site-packages/google/cloud/bigquery/table.py:1933:
UserWarning: BigQuery Storage module not found, fetch data with the REST end
point instead.
  warnings.warn(
```

In [109…

```
lab_query = """
SELECT le.subject_id, le.hadm_id, le.charttime, d.label AS lab_name, le.valu
FROM `physionet-data.mimiciii_clinical.labevents` le
JOIN `physionet-data.mimiciii_clinical.d_labitems` d
  ON le.itemid = d.itemid
WHERE le.hadm_id IN {hadm_filter}
AND le.valuenum IS NOT NULL

"""

lab_df = client.query(lab_query.format(hadm_filter=hadm_filter)).to_datafram
lab_df.to_csv("./neo4j-community-2025.03.0/import/labevents.csv", index=Fals
```

```
/Users/abemankavil/Desktop/MS in AI/AI395T-AIinHealthcare/HighRisk_Project/H
ighRiskEnv/lib/python3.9/site-packages/google/cloud/bigquery/table.py:1933:
UserWarning: BigQuery Storage module not found, fetch data with the REST end
point instead.
  warnings.warn(
```

In [1]:

```
from neo4j import GraphDatabase
from langchain.graphs import Neo4jGraph
from langchain.chat_models import ChatOpenAI
from langchain.chains import GraphCypherQAChain
from sklearn.ensemble import IsolationForest
import numpy as np
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt
import pandas as pd
from openai import OpenAI
from graphdatascience import GraphDataScience
import seaborn as sns
```

```
/Users/abemankavil/Desktop/MS in AI/AI395T-AIinHealthcare/EHR_Anomaly_Detect
ion/HighRiskEnv/lib/python3.9/site-packages/tqdm/auto.py:21: TqdmWarning: IP
rogress not found. Please update jupyter and ipywidgets. See https://ipywidg
ets.readthedocs.io/en/stable/user_install.html
  from .autonotebook import tqdm as notebook_tqdm
```

In [ ]:

```
# -------------------------------
# Step 1: Neo4j + OpenAI Setup
# -------------------------------

NEO4J_URL = "bolt://localhost:7687"


# LangChain graph wrapper
graph_langchain = Neo4jGraph(
    url=NEO4J_URL,
    username=NEO4J_USER,
    password=NEO4J_PASSWORD
)

# Neo4j driver and GDS interface
gds = GraphDataScience(NEO4J_URL, auth=(NEO4J_USER, NEO4J_PASSWORD))
```

```
/var/folders/0r/lprjchvd593dyrpzsrg6jjrm0000gn/T/ipykernel_83891/3600210996.
py:13: LangChainDeprecationWarning: The class `Neo4jGraph` was deprecated in
LangChain 0.3.8 and will be removed in 1.0. An updated version of the class
exists in the :class:`~langchain-neo4j package and should be used instead. T
o use it run `pip install -U :class:`~langchain-neo4j` and import as `from :
class:`~langchain_neo4j import Neo4jGraph``.
  graph_langchain = Neo4jGraph(
```

In [4]:
```
# -------------------------------
# Step 2: Generate Embeddings Using GDS Python Client
# -------------------------------

def generate_node2vec_embeddings():
    if gds.graph.exists("admission_subgraph").get("exists", False):
        gds.graph.drop("admission_subgraph")

    G, result = gds.graph.project(
        "admission_subgraph",
        node_spec={"Admission": {},
                    "Diagnosis": {},
                    "LabResult": {},
                    "Medication": {}},
        relationship_spec={
            "HAS_DIAGNOSIS": {"orientation": "UNDIRECTED"},
            "HAS_LAB": {"orientation": "UNDIRECTED"},
            "HAS_MEDICATION": {"orientation": "UNDIRECTED"}
        }
    )

    print(f"Projection created in {result['projectMillis']} ms")

    result = gds.node2vec.write(
    G,  # Graph object from projection
    writeProperty="embedding",
    embeddingDimension=64,
    iterations=10,
    nodeLabels=["Admission"],
    concurrency=4
    )
    print(f"Node2Vec computed for {result['nodeCount']} nodes")
```

In [5]:
```
# -------------------------------
# Step 3: Fetch Embeddings from Neo4j
# -------------------------------

def get_admission_features():
    query = """
    MATCH (p:Patient)-[:HAS_ADMISSION]->(a:Admission)
    OPTIONAL MATCH (a)-[:HAS_MEDICATION]->(m:Medication)
    OPTIONAL MATCH (a)-[:HAS_LAB]->(l:LabResult)
    OPTIONAL MATCH (a)-[:HAS_DIAGNOSIS]->(d:Diagnosis)
    WHERE a.embedding IS NOT NULL
    RETURN a.HADM_ID AS hadm_id, a.embedding AS embedding, a.admission_type
           p.GENDER AS gender, a.ethnicity AS ethnicity, a.age AS age, a.hos
           collect(DISTINCT {name: m.DRUG, dose: m.dose, unit: m.unit}) AS m
    """
    result = gds.run_cypher(query)
    df = pd.DataFrame(result)
    df = df[df["embedding"].notnull()].copy()
    df["embedding"] = df["embedding"].apply(lambda x: [float(i) for i in x])
    return df
```

In [98]:

```
# ------------------------------
# Step 4: Detect Anomalies Using Isolation Forest
# ------------------------------
from sklearn.preprocessing import OneHotEncoder, MultiLabelBinarizer

def detect_anomalies(df):
    cat_features = df[["gender", "ethnicity", "admission_type", "hospital_ex
    enc = OneHotEncoder(sparse_output=False, handle_unknown="ignore")
    cat_encoded = enc.fit_transform(cat_features)

    meds_name_enc = MultiLabelBinarizer()
    meds_dose_enc = MultiLabelBinarizer()
    labs_name_enc = MultiLabelBinarizer()
    labs_val_enc = MultiLabelBinarizer()
    diag_enc = MultiLabelBinarizer()

    # Split medications into name and (dose+unit)
    def split_medications(meds):
        names = [m["name"].lower() for m in meds if m.get("name")]
        doses = [f"{m.get('dose', '?')} {m.get('unit', '?')}" for m in meds
        return pd.Series([names, doses])


    # Split labs into name and (value+unit)
    def split_labs(labs):
        names = [l["name"].lower() for l in labs if l.get("name")]
        values = [f"{l.get('value', '?')} {l.get('unit', '')}" for l in labs
        return pd.Series([names, values])

    df[["med_names", "med_doses"]] = df["meds"].apply(split_medications)
    df[["lab_names", "lab_values"]] = df["labs"].apply(split_labs)


    med_name_feats = meds_name_enc.fit_transform(df["med_names"])
    med_dose_feats = meds_dose_enc.fit_transform(df["med_doses"])
    lab_name_feats = labs_name_enc.fit_transform(df["lab_names"])
    lab_val_feats = labs_val_enc.fit_transform(df["lab_values"])
    diag_feats = diag_enc.fit_transform(df["diag_codes"])

    age_vals = df["age"].fillna(0).values.reshape(-1, 1)
    embed_vals = np.vstack(df["embedding"].values)
    X = np.hstack([embed_vals, age_vals, cat_encoded, med_dose_feats, med_na

    model = IsolationForest(n_estimators=100, contamination=0.05, random_sta
    df['anomaly_score'] = model.fit_predict(X)
    df['anomaly'] = df['anomaly_score'] == -1
    df['score'] = model.decision_function(X)

    X_tsne = TSNE(n_components=2, perplexity=30, learning_rate=200, n_iter=1

    # Plot
    plt.figure(figsize=(10, 6))
    scatter = plt.scatter(
        X_tsne[:, 0], X_tsne[:, 1],
        c=df["anomaly"],
        cmap="coolwarm",
        alpha=0.7,
        edgecolor='k',
        linewidth=0.3
    )

    plt.title("t-SNE Projection of EHR Admissions (Anomalies Highlighted)")
    plt.xlabel("t-SNE Dimension 1")
    plt.ylabel("t-SNE Dimension 2")
    plt.legend(*scatter.legend_elements(), title="Anomaly = 1", loc="lower r
    plt.grid(True, linestyle="--", alpha=0.3)
    plt.tight_layout()
    plt.show()

    plt.figure(figsize=(10, 6))
    sns.histplot(df["score"], bins=50, kde=True)
    plt.axvline(df[df['anomaly'] == True]['score'].mean(), color='red', line
    plt.title("Distribution of Anomaly Scores (Isolation Forest)")
    plt.xlabel("Anomaly Score")
    plt.ylabel("Frequency")
    plt.legend()
    plt.tight_layout()
    plt.show()



    return df[df['anomaly'] == True].sort_values("score")
```

In [ ]:

```
# -------------------------------
# Step 5: LLM-Based Explanation via LangChain
# -------------------------------

llm = ChatOpenAI(
model="gpt-4o-mini",
temperature=0.3,
openai_api_key=openai_api_key
)

# GraphCypherQAChain Setup
qa_chain = GraphCypherQAChain.from_llm(
    llm=llm,
    graph=graph_langchain,
    verbose=False,
    allow_dangerous_requests=True
)

def explain_anomaly(hadm_id):
    question = f"""

You are a medical graph reasoning assistant. A patient admission is being ev

Only use the following information in your Cypher query:
- Admission attributes: admission_type, diagnosis_free_text, age, ethnicity,
- Patient attributes: GENDER
- Diagnosis: CODE
- LabResult: TEST_NAME, value, unit
- Medication: DRUG, dose, unit


Ensure you **aggregate and deduplicate** all relevant values using `COLLECT(
- Medications: `collect(DISTINCT {{name: m.DRUG, dose: m.dose, unit: m.unit}
- Lab Results: `collect(DISTINCT {{name: l.TEST_NAME, value: l.value, unit:
- Diagnosis: `collect(DISTINCT d.CODE) AS diag_codes`


Explain why the hospital admission with HADM_ID {hadm_id} might be clinicall
Identify 2 to 3 clinically meaningful anomalies for this hospital admission.

Your analysis can consider, but is not limited to, any of the following:
• Unusual medication-diagnosis combinations
• Rare or unexpected lab test patterns based on admission type
• Mismatches between patient demographics (e.g., age, gender, ethnicity) anc
• Co-occurrences of diagnoses that are atypical when found together

You may also identify other patterns or inconsistencies you find clinically

"""
    try:
        return qa_chain.invoke(question)['result']
    except Exception as e:
        return f"LLM explanation failed: {e}"
```

In [8]:
```
# -------------------------------
# Step 6: Run Full Pipeline
# -------------------------------

print("[1] Generating Node2Vec embeddings using GDS Python client...")
generate_node2vec_embeddings()
```

```
[1] Generating Node2Vec embeddings using GDS Python client...
Projection created in 816 ms
Node2Vec computed for 1163 nodes
```

In [9]:
```
print("[2] Fetching embeddings + node properties from Neo4j...")
df = get_admission_features()
```

```
[2] Fetching embeddings + node properties from Neo4j...
```

```
/Users/abemankavil/Desktop/MS in AI/AI395T-AIinHealthcare/HighRisk_Project/HighRiskEnv/lib/python3.9/site-packages/graphdatascience/query_runner/neo4j_query_runner.py:277: RuntimeWarning: null value eliminated in set function.
  warnings.warn(warning)
```

In [77]:
```
df.head()
```

Out[77]:

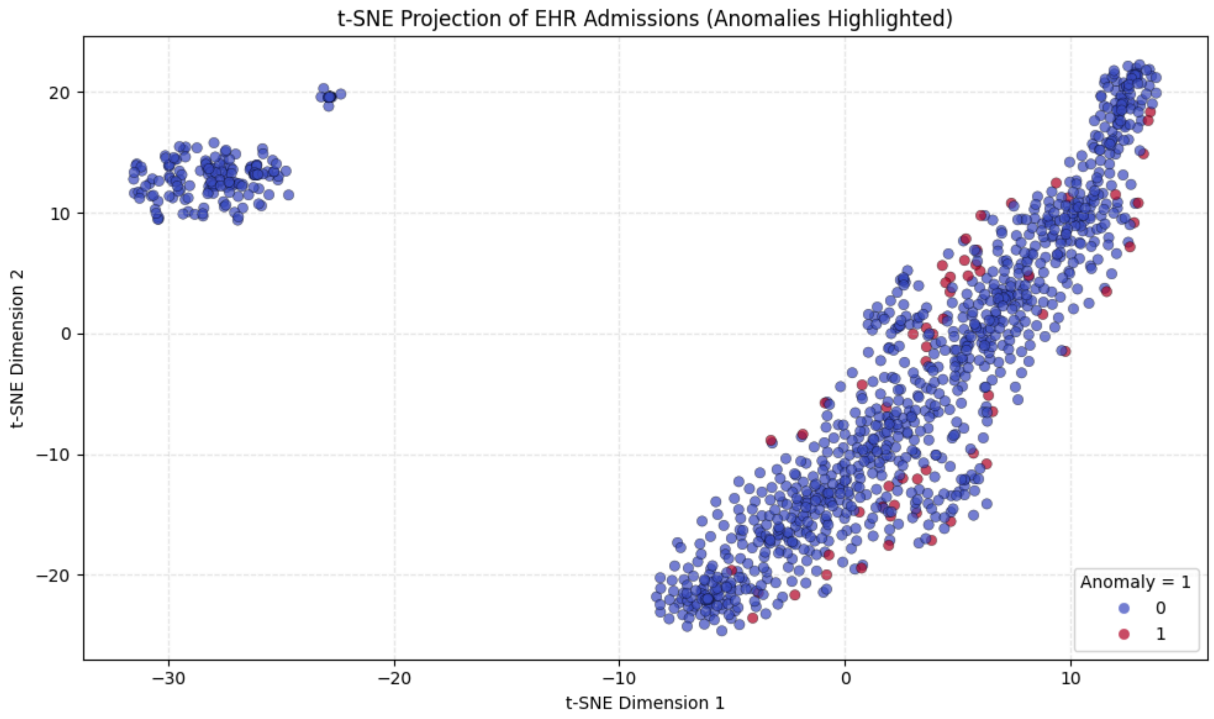
	hadm_id	embedding	admission_type	diagnosis	gender
0	144471	[-0.0001035963068716228, -0.007562919519841671...	EMERGENCY	WEAKNESS	F
1	162937	[-0.0008790345746092498, 0.0069307610392570496...	EMERGENCY	MULTI MYELOMA;PAIN	F
2	169482	[-0.0008776350878179073, 0.004281629808247089,...	EMERGENCY	ACUTE GASTROENTERISTIS	F
3	177186	[0.0010735585819929838, -0.0011072519700974226...	EMERGENCY	ALTERED MENTAL STATUS	F
4	127794	[0.0007670226041227579, 0.0009274794138036668,...	ELECTIVE	ANEURYSM/SDA	F

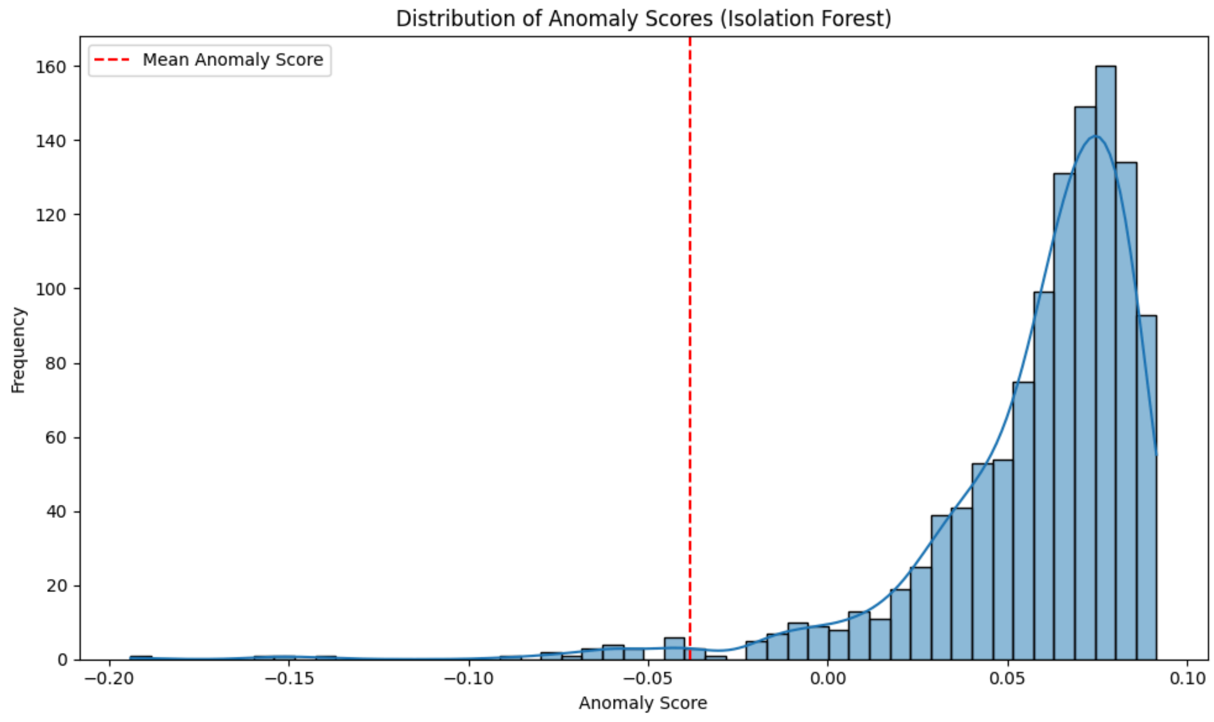
5 rows × 21 columns

In [99]:
```
print("[3] Running Isolation Forest to detect anomalies...")
anomalies = detect_anomalies(df)
print(f"Detected {len(anomalies)} anomalous admissions.\n")
```

```
[3] Running Isolation Forest to detect anomalies...
```

```
/Users/abemankavil/Desktop/MS in AI/AI395T-AIinHealthcare/HighRisk_Project/H
ighRiskEnv/lib/python3.9/site-packages/sklearn/manifold/_t_sne.py:1164: Futu
reWarning: 'n_iter' was renamed to 'max_iter' in version 1.5 and will be rem
oved in 1.7.
  warnings.warn(
```





```
Detected 59 anomalous admissions.
```

In [25]:
```python
import concurrent.futures

def explain_anomaly_wrapper(hadm_id):
    try:
        explanation = explain_anomaly(hadm_id)
        return hadm_id, explanation
    except Exception as e:
        return hadm_id, f"Error: {e}"

N = int(input("How many top-N anomalies would you like to explain? "))
top_anomalies = anomalies.head(N)

print("[4] Generating explanations for top anomalies...\n")

with concurrent.futures.ThreadPoolExecutor() as executor:
    futures = {executor.submit(explain_anomaly_wrapper, row['hadm_id']): row
    for future in concurrent.futures.as_completed(futures):
        hadm_id, explanation = future.result(timeout=90)
        print(f"HADM_ID: {hadm_id}")
        print("Explanation:", explanation)
        print("-" * 60)
```

```
[4] Generating explanations for top anomalies...

HADM_ID: 164694
Explanation: The hospital admission with HADM_ID 164694 may be clinically an
omalous for several reasons:

1. **Diagnosis and Age Mismatch**: The patient is a 38-year-old male diagnos
ed with Acute Myelogenous Leukemia (AML), which is more commonly diagnosed i
n older adults. This diagnosis at a relatively young age could be considered
unusual and warrants further investigation into potential underlying genetic
or environmental factors.

2. **Medication-Diagnosis Combination**: The patient is receiving a wide arr
ay of medications, including Quetiapine Fumarate, which is primarily used fo
r psychiatric conditions. This raises concerns about the appropriateness of
this medication in the context of AML treatment, as it may not be a standard
part of the therapeutic regimen for this diagnosis.

3. **Lab Test Patterns**: The lab results show a notably low platelet count
(29.0 K/uL), which is expected in patients with leukemia. However, the prese
nce of atypical lymphocytes (10.0%) and a high lactate dehydrogenase (LD) le
vel (1168.0 IU/L) could indicate a more severe underlying condition or compl
ications, such as tumor lysis syndrome or an infection, which are concerning
in the context of AML.

These anomalies suggest that the patient's clinical presentation may not ali
gn with typical expectations for their diagnosis and demographic profile, in
dicating a need for closer scrutiny and possibly a reevaluation of their tre
atment plan.
------------------------------------------------------------
HADM_ID: 119862
Explanation: The hospital admission with HADM_ID 119862 may be clinically an
omalous for several reasons:

1. **Unusual Medication-Diagnosis Combinations**: The patient, a 57-year-old
Black/Haitian male admitted for fever under emergency conditions, received a
wide array of medications that may not typically be associated with a straig
htforward fever diagnosis. For instance, the administration of medications s
uch as Thrombin and Filgrastim suggests a potential underlying condition tha
t requires hemostatic support or stimulation of bone marrow activity, which
is not common for a fever alone.

2. **Rare Lab Test Patterns**: The lab results indicate significant abnormal
ities, such as a high level of Alanine Aminotransferase (ALT) at 940 IU/L, w
hich is markedly elevated and may suggest liver dysfunction or damage. Addit
ionally, the presence of elevated lactate levels (up to 12.0 mmol/L) could i
ndicate tissue hypoperfusion or sepsis, which aligns with the emergency admi
ssion but raises concerns about the patient's overall stability and potentia
l underlying conditions.

3. **Mismatches Between Patient Demographics and Clinical Presentation**: Th
e patient's age and ethnicity may also present a clinical anomaly when consi
dering the typical causes of fever in this demographic. For instance, certai
n infections or conditions that commonly cause fever may be less prevalent o
r present atypically in older Black/Haitian males, suggesting a need for fur
ther investigation into the underlying causes of the fever.
```

```
These anomalies warrant further clinical evaluation to ensure appropriate di
agnosis and treatment, as they may indicate more complex underlying health i
ssues.
------------------------------------------------------------
HADM_ID: 131328
Explanation: The hospital admission with HADM_ID 131328 may be clinically an
omalous for several reasons:

1. **Unusual Medication-Diagnosis Combinations**: The patient, a 53-year-old
white female diagnosed with acute myeloid leukemia and undergoing a bone mar
row transplant, is prescribed a wide array of medications, including high do
ses of chemotherapeutic agents like Cyclophosphamide (4860 mg) and Mitoxantr
one (15 mg). Such high dosages, particularly in combination with other medic
ations like Fentanyl Citrate (2.5 mg) and Morphine Sulfate (15 mg), raise co
ncerns about potential drug interactions and the management of pain and side
effects during a critical treatment phase.

2. **Rare Lab Test Patterns**: The lab results show a significant variation
in key indicators. For instance, the patient's creatinine levels fluctuate b
etween 0.4 mg/dL and 2.0 mg/dL, indicating possible acute kidney injury or r
enal impairment, which is not uncommon in patients undergoing chemotherapy.
Additionally, the presence of atypical lymphocytes (0.0% to 1.0%) alongside
elevated levels of lactate dehydrogenase (up to 1290 IU/L) may suggest under
lying complications or an unusual response to treatment.

3. **Demographic Mismatches**: The patient's age and gender may also present
anomalies. While acute myeloid leukemia can occur in older adults, the combi
nation of a relatively young female patient with such a severe diagnosis and
the aggressive treatment protocol raises questions about the appropriateness
of the treatment plan, especially considering the potential for adverse effe
cts in this demographic.

These factors collectively suggest that the admission may involve complexiti
es that warrant further investigation to ensure optimal patient care and man
agement.
------------------------------------------------------------
```