In [16]:
```
import pandas as pd
import numpy as np
from datetime import datetime
import re
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import jaccard_score
from Levenshtein import distance as lev_distance
```

In [17]:
```
# Step 2: Load CSV Files
merged_cassandra = pd.read_csv('merged/merged_cassandra.csv')
merged_mongo = pd.read_csv('merged/merged_mongo.csv')
merged_postgres = pd.read_csv('merged/merged_postgres.csv')
merged_web = pd.read_csv('merged/merged_web.csv')
merged_xml = pd.read_csv('merged/merged_xml.csv')
merged_mysql = pd.read_csv('merged/merged_mysql.csv')

# Display the first few rows of each dataset to get an idea of the data structure
merged_cassandra.head(), merged_mongo.head(), merged_postgres.head(
), merged_web.head(), merged_xml.head(), merged_mysql.head()
```

Out[17]: (

	ClientID	ClientName	Address	ContactNumber	AreaType	AreaDescription	Classification	ClassificationDescription	AccountStatus	StatusDescription	ContractID	StartDate	EndDate	BaseFee	LastUpdated	ContractStatus	ContractStatusDescription	PolicyName	PolicyDetails	EffectiveDate
0	ea614ade-9cee-43ba-bb90-319f7079f8dc	Customer_4	Rua 4, Faro	35191000004	Residential	Residential area for housing	General	General level clients	No Breach	Client is in good standing	15b3f64a-d57e-4517-8e48-8faef8724db1	2024-01-26	2026-01-25	100.0	2025-01-25 17:03:24.432000	Terminated	The contract has been terminated	Policy 3	Details of policy 3	2025-01-25
1	ea614ade-9cee-43ba-bb90-319f7079f8dc	Customer_4	Rua 4, Faro	35191000004	Residential	Residential area for housing	General	General level clients	No Breach	Client is in good standing	15b3f64a-d57e-4517-8e48-8faef8724db1	2024-01-26	2026-01-25	100.0	2025-01-25 17:03:24.432000	Terminated	The contract has been terminated	Policy 1	Details of policy 1	2025-01-25
2	ea614ade-9cee-43ba-bb90-319f7079f8dc	Customer_4	Rua 4, Faro	35191000004	Residential	Residential area for housing	General	General level clients	No Breach	Client is in good standing	15b3f64a-d57e-4517-8e48-8faef8724db1	2024-01-26	2026-01-25	100.0	2025-01-25 17:03:24.432000	Terminated	The contract has been terminated	Policy 2	Details of policy 2	2025-01-25
3	73688418-9121-4b41-bc92-0a14a9b060ed	Customer_127	Rua 127, Faro	35192000060	Industrial	Industrial zones with factories	Premium	Premium level clients	Compliance Issue	Client has compliance-related issues	92dcc6ec-9442-415d-95c3-951a5572e926	2024-01-26	2026-01-25	100.0	2025-01-25 17:03:24.432000	Suspended	The contract is temporarily suspended	Policy 3	Details of policy 3	2025-01-25
4	73688418-9121-4b41-bc92-0a14a9b060ed	Customer_127	Rua 127, Faro	35192000060	Industrial	Industrial zones with factories	Premium	Premium level clients	Compliance Issue	Client has compliance-related issues	92dcc6ec-9442-415d-95c3-951a5572e926	2024-01-26	2026-01-25	100.0	2025-01-25 17:03:24.432000	Suspended	The contract is temporarily suspended	Policy 1	Details of policy 1	2025-01-25

,

	_id	Name	ConsumerType	Address	ContactInfo	RelevanceID	ContractID	ContractStartDate
0	6794b76b1d956a8a4af23e88	Customer_1	Residential	rua 1, Faro				
1	6794b76b1d956a8a4af23e89	Customer_2	Residential	rua 2, Faro				
2	6794b76b1d956a8a4af23e8a	Customer_3	Residential	rua 3, Faro				
3	6794b76b1d956a8a4af23e8b	Customer_4	Residential	rua 4, Faro				
4	6794b76b1d956a8a4af23e8c	Customer_5	Residential	rua 5, Faro				

```
0  35191000001  6794b7501d956a8a4af23e83    CT00001        2023-01-11
1  35191000002  6794b7511d956a8a4af23e84    CT00002        2023-01-21
2  35191000003  6794b7501d956a8a4af23e82    CT00003        2023-01-31
3  35191000004  6794b7501d956a8a4af23e83    CT00004        2023-02-10
4  35191000005  6794b7511d956a8a4af23e84    CT00005        2023-02-20

  ContractEndDate ContractCommitment  ... ProgramName  \
0      2025-01-11         2024-01-06  ...   Program_1
1      2025-01-21         2024-01-11  ...   Program_2
2      2025-01-31         2024-01-16  ...   Program_3
3      2025-02-10         2024-01-21  ...   Program_4
4      2025-02-20         2024-01-26  ...   Program_5

  TypeID_WaterConservationPrograms StartDate_WaterConservationPrograms  \
0         6794b7511d956a8a4af23e86                           2023-01-31
1         6794b7511d956a8a4af23e87                           2023-03-02
2         6794b7511d956a8a4af23e85                           2023-04-01
3         6794b7511d956a8a4af23e86                           2023-05-01
4         6794b7511d956a8a4af23e87                           2023-05-31

     EndDate TargetArea _id_ConservationParticipants  \
0  2025-01-31   Region_1     6794b7a21d956a8a4af24126
1  2025-03-02   Region_2     6794b7a21d956a8a4af24127
2  2025-04-01   Region_3     6794b7a21d956a8a4af24128
3  2025-05-01   Region_4     6794b7a21d956a8a4af24129
4  2025-05-31   Region_5     6794b7a21d956a8a4af2412a

  CustomerID_ConservationParticipants  ProgramID ParticipationStartDate  \
0                                   1          2             2023-01-03
1                                   2          3             2023-01-05
2                                   3          4             2023-01-07
3                                   4          5             2023-01-09
4                                   5          6             2023-01-11

                     BenefitsReceived
0  Benefits for Customer 1 in Program 2
1  Benefits for Customer 2 in Program 3
2  Benefits for Customer 3 in Program 4
3  Benefits for Customer 4 in Program 5
4  Benefits for Customer 5 in Program 6

[5 rows x 35 columns],
   ClientID        Name  FullAddress   Postcode   MobileInfo     AreaType  \
0         1  Customer_1  rua 1, Faro  801-00001  35191000001  Residential
1         1  Customer_1  rua 1, Faro  801-00001  35191000001  Residential
2         1  Customer_1  rua 1, Faro  801-00001  35191000001  Residential
3         1  Customer_1  rua 1, Faro  801-00001  35191000001  Residential
4         1  Customer_1  rua 1, Faro  801-00001  35191000001  Residential

  CollectionFrequency LastCollectionDate NextCollectionDate  WasteType  \
0               Daily         2025-01-24         2025-01-25    Organic
1               Daily         2025-01-24         2025-01-25    Organic
2               Daily         2025-01-24         2025-01-25    Organic
3               Daily         2025-01-24         2025-01-25    Organic
4               Daily         2025-01-24         2025-01-25  Hazardous
```

```
   UnitPricePerKg DisposalDate  QuantityInKg BillingDate  TotalAmount  \
0             0.5   2025-01-25           0.1  2025-01-24          5.0
1             0.5   2025-01-25           0.1  2025-01-24          5.0
2             0.5   2025-01-25           0.1  2025-01-24          5.0
3             0.5   2025-01-25           0.1  2025-01-24          5.0
4             1.5   2025-01-24           0.2  2025-01-24          5.0

   BillingQuantity  SubTotal
0             50.0      25.0
1             50.0      25.0
2             50.0      25.0
3             50.0      25.0
4             50.0      25.0  ,
   ReportID  Validation                 Time  ReservoirID  Temperature  \
0         1           0  2024-01-15T00:00:00            1        26.86
1         2           0  2024-08-27T00:00:00            2        10.25
2         3           0  2024-08-27T00:00:00            2        10.25
3         4           1  2024-02-07T00:00:00            4        34.93
4         5           1  2024-04-20T00:00:00            5        31.50

   Turbidity    pH     ReportURL  SensorID SensorType     Location  \
0       5.30  7.42  report_1.pdf        52  Turbidity  Reservoir_5
1       5.68  8.40  report_2.pdf        25  Turbidity  Reservoir_3
2       5.68  8.40  report_3.pdf        25  Turbidity  Reservoir_3
3       2.67  7.70  report_4.pdf         8         pH  Reservoir_5
4       4.32  6.84  report_5.pdf        62         pH  Reservoir_1

      InstallationDate  Status  pHDataID  Report_ReportID  TempRecordDataID  \
0  2021-03-29T00:00:00       0         5              482                 6
1  2021-12-05T00:00:00       0         3              614                 1
2  2021-12-05T00:00:00       0         3              614                 1
3  2022-05-17T00:00:00       1         1              152                 4
4  2023-07-24T00:00:00       1         7              326                 3

   TurbidityDataID
0                6
1                2
2                2
3                1
4                3  ,
  ClientID        Name      Contact      Address     AreaType ComplaintID  \
0       C2  Customer_2  35191000002  Rua 2, Faro  Residential       COMP1
1       C3  Customer_3  35191000003  Rua 3, Faro  Residential       COMP2
2       C4  Customer_4  35191000004  Rua 4, Faro  Residential       COMP3
3       C5  Customer_5  35191000005  Rua 5, Faro  Residential       COMP4
4       C6  Customer_6  35191000006  Rua 6, Faro  Residential       COMP5

             Type        Date        Status IssueID IssueType BreachID Reason
0  Contract Breach  2025-01-23       Pending     NaN       NaN      NaN    NaN
1         Renewal  2025-01-22       Pending     NaN       NaN      NaN    NaN
2         Billing  2025-01-21       Pending     NaN       NaN      NaN    NaN
3  Contract Breach  2025-01-20       Pending     NaN       NaN      NaN    NaN
4         Renewal  2025-01-19  Under Review     NaN       NaN      NaN    NaN  ,
   programid programname  programtypeid   startdate     enddate description  \
0          1   Program 1              1  2024-01-01  2025-01-01   Education
1          1   Program 1              1  2024-01-01  2025-01-01   Education
```

```
2          1   Program 1              1  2024-01-01  2025-01-01   Education
3          1   Program 1              1  2024-01-01  2025-01-01   Education
4          1   Program 1              1  2024-01-01  2025-01-01   Education

   usageid  departmentid_x  sourceid   usagedate  ...  testid    testdate  \
0        1               1         1  2020-01-28  ...       1  2020-01-28
1        1               1         1  2020-01-28  ...       6  2020-02-02
2        1               1         1  2020-01-28  ...      11  2020-02-07
3        1               1         1  2020-01-28  ...      16  2020-02-12
4        1               1         1  2020-01-28  ...      21  2020-02-17

     testresults participationid customerid participationstartdate  \
0  Test result 2               1          1             2024-01-01
1  Test result 2               1          1             2024-01-01
2  Test result 2               1          1             2024-01-01
3  Test result 2               1          1             2024-01-01
4  Test result 2               1          1             2024-01-01

         name  consumertype          address   contactinfo
0  Customer_1   Residential  Rua 1, CityZone  351910000001
1  Customer_1   Residential  Rua 1, CityZone  351910000001
2  Customer_1   Residential  Rua 1, CityZone  351910000001
3  Customer_1   Residential  Rua 1, CityZone  351910000001
4  Customer_1   Residential  Rua 1, CityZone  351910000001

[5 rows x 34 columns])
```

In [18]:

```python
def normalize_dates(df, date_columns):
    for col in date_columns:
        if col in df.columns:
            df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
    return df


# Define the date columns for each dataset (you can customize these as needed)
date_columns_cassandra = ['StartDate',
                          'EndDate', 'EffectiveDate', 'LastUpdated']
date_columns_mongo = ['ContractStartDate',
                      'ContractEndDate', 'PublicationDate', 'EffectiveDate']
date_columns_postgres = ['LastCollectionDate',
                         'NextCollectionDate', 'DisposalDate', 'BillingDate']
date_columns_web = ['Time', 'InstallationDate']
date_columns_xml = ['Date']
date_columns_mysql = ['startdate', 'enddate', 'usagedate',
                      'participationstartdate', 'lastinspectiondate', 'maintenanced

# Apply normalization
merged_cassandra = normalize_dates(merged_cassandra, date_columns_cassandra)
merged_mongo = normalize_dates(merged_mongo, date_columns_mongo)
merged_postgres = normalize_dates(merged_postgres, date_columns_postgres)
merged_web = normalize_dates(merged_web, date_columns_web)
merged_xml = normalize_dates(merged_xml, date_columns_xml)
merged_mysql = normalize_dates(merged_mysql, date_columns_mysql)
```

```
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d %H:%M:%S.%f format when dayfirst=True was specified. Pass `day
first=False` or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%dT%H:%M:%S format when dayfirst=True was specified. Pass `dayfir
st=False` or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%dT%H:%M:%S format when dayfirst=True was specified. Pass `dayfir
st=False` or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
```

```
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3012549597.py:4: UserWarning: Pars
ing dates in %Y-%m-%d format when dayfirst=True was specified. Pass `dayfirst=False`
or specify a format to silence this warning.
  df[col] = pd.to_datetime(df[col], errors='coerce', dayfirst=True)
```

In [19]:
```python
# Step 4: Normalize Contact Information
def normalize_contacts(df, contact_columns):
    for col in contact_columns:
        if col in df.columns:
            # Ensure contact is treated as a string
            df[col] = df[col].astype(str)
    return df


# Define the contact columns for each dataset
contact_columns_cassandra = ['ContactNumber']
contact_columns_mongo = ['ContactInfo']
contact_columns_postgres = ['MobileInfo']
# If sensor IDs need normalization, adjust as needed
contact_columns_web = ['SensorID']
contact_columns_xml = ['Contact']
contact_columns_mysql = ['contactinfo']

# Apply normalization
merged_cassandra = normalize_contacts(
    merged_cassandra, contact_columns_cassandra)
merged_mongo = normalize_contacts(merged_mongo, contact_columns_mongo)
merged_postgres = normalize_contacts(merged_postgres, contact_columns_postgres)
merged_web = normalize_contacts(merged_web, contact_columns_web)
merged_xml = normalize_contacts(merged_xml, contact_columns_xml)
merged_mysql = normalize_contacts(merged_mysql, contact_columns_mysql)
```

In [20]:
```python
# Step 5: Normalize Other Columns (Optional)
def normalize_numerical(df, numerical_columns):
    for col in numerical_columns:
        if col in df.columns:
            df[col] = pd.to_numeric(df[col], errors='coerce')
    return df


# Example numerical columns (you can adjust this list based on your data)
numerical_columns = ['BaseFee', 'TotalAmount',
                     'QuantityInKg', 'VolumeUsed', 'ProjectedSavings']

# Apply normalization to numerical columns
merged_cassandra = normalize_numerical(merged_cassandra, numerical_columns)
merged_mongo = normalize_numerical(merged_mongo, numerical_columns)
merged_postgres = normalize_numerical(merged_postgres, numerical_columns)
merged_web = normalize_numerical(merged_web, numerical_columns)
merged_xml = normalize_numerical(merged_xml, numerical_columns)
merged_mysql = normalize_numerical(merged_mysql, numerical_columns)
```

In [21]:
```python
# Step 6: Jaccard Similarity on Column Names
def jaccard_similarity(list1, list2):
    set1, set2 = set(list1), set(list2)
    return len(set1.intersection(set2)) / len(set1.union(set2))
```

```
# Get the column names for each dataset
columns_cassandra = merged_cassandra.columns.tolist()
columns_mongo = merged_mongo.columns.tolist()
columns_postgres = merged_postgres.columns.tolist()
columns_web = merged_web.columns.tolist()
columns_xml = merged_xml.columns.tolist()
columns_mysql = merged_mysql.columns.tolist()

# Compute the Jaccard similarity between column names
jaccard_cassandra_mongo = jaccard_similarity(columns_cassandra, columns_mongo)
jaccard_cassandra_postgres = jaccard_similarity(
    columns_cassandra, columns_postgres)
jaccard_cassandra_web = jaccard_similarity(columns_cassandra, columns_web)
jaccard_cassandra_xml = jaccard_similarity(columns_cassandra, columns_xml)
jaccard_cassandra_mysql = jaccard_similarity(columns_cassandra, columns_mysql)

# Display results
print("Jaccard Similarity Scores:")
print(f"Cassandra vs Mongo: {jaccard_cassandra_mongo}")
print(f"Cassandra vs Postgres: {jaccard_cassandra_postgres}")
print(f"Cassandra vs Web: {jaccard_cassandra_web}")
print(f"Cassandra vs XML: {jaccard_cassandra_xml}")
print(f"Cassandra vs MySQL: {jaccard_cassandra_mysql}")
```

```
Jaccard Similarity Scores:
Cassandra vs Mongo: 0.12244897959183673
Cassandra vs Postgres: 0.05714285714285714
Cassandra vs Web: 0.0
Cassandra vs XML: 0.1
Cassandra vs MySQL: 0.0
```

In [22]:

```
# Step 7: Levenshtein Similarity on Column Names
def levenshtein_similarity(str1, str2):
    return 1 - (lev_distance(str1, str2) / max(len(str1), len(str2)))


# Compute Levenshtein similarity for some example column pairs
levenshtein_cassandra_mongo = [levenshtein_similarity(
    c, m) for c in columns_cassandra for m in columns_mongo]
levenshtein_cassandra_postgres = [levenshtein_similarity(
    c, p) for c in columns_cassandra for p in columns_postgres]

# Example results for a few column names
print("Levenshtein Similarity Scores:")
print(f"Levenshtein Scores between Cassandra and Mongo columns: {levenshtein_cassan
print(f"Levenshtein Scores between Cassandra and Postgres columns: {levenshtein_cas
```

```
Levenshtein Similarity Scores:
Levenshtein Scores between Cassandra and Mongo columns: [0.125, 0.125, 0.16666666666
666663, 0.0, 0.2727272727272727, 0.36363636363636365, 0.4, 0.17647058823529416, 0.19
999999999999996, 0.16666666666666663]
Levenshtein Scores between Cassandra and Postgres columns: [1.0, 0.125, 0.0909090909
0909094, 0.0, 0.19999999999999996, 0.0, 0.21052631578947367, 0.2777777777777778, 0.2
777777777777778, 0.11111111111111116]
```

In [23]:
```python
# Step 8: Analyze and Integrate Best Matches (Example)

# Create a dictionary to store the best matches for each pair of datasets
best_matches = {
    'Cassandra_Mongo': (jaccard_cassandra_mongo, levenshtein_cassandra_mongo),
    'Cassandra_Postgres': (jaccard_cassandra_postgres, levenshtein_cassandra_postgr
}

# Print the best matches
for pair, scores in best_matches.items():
    print(f"Best matches for {pair}:")
    print(f"Jaccard Similarity: {scores[0]}")
    print(f"Levenshtein Similarity: {scores[1]}")
```

```
Best matches for Cassandra_Mongo:
Jaccard Similarity: 0.12244897959183673
Levenshtein Similarity: [0.125, 0.125, 0.16666666666666663, 0.0, 0.2727272727272727,
0.36363636363636365, 0.4, 0.17647058823529416, 0.19999999999999996, 0.16666666666666
663, 0.375, 0.21052631578947367, 0.19999999999999996, 0.19999999999999996, 0.2666666
666666667, 0.23076923076923073, 0.07692307692307687, 0.30000000000000004, 0.375, 0.0
, 0.125, 0.14814814814814814, 0.25, 0.2857142857142857, 0.10344827586206895, 0.0, 0.
09375, 0.08571428571428574, 0.125, 0.09999999999999998, 0.1428571428571429, 0.114285
71428571432, 0.2222222222222222, 0.18181818181818177, 0.125, 0.09999999999999998, 0.
4, 0.16666666666666663, 0.0, 0.18181818181818177, 0.09090909090909094, 0.19999999999
999996, 0.23529411764705888, 0.2666666666666667, 0.2222222222222222, 0.3000000000000
0004, 0.21052631578947367, 0.5, 0.09999999999999998, 0.33333333333333337, 0.30769230
76923077, 0.11538461538461542, 0.09999999999999998, 0.09999999999999998, 0.300000000
00000004, 0.1875, 0.18518518518518523, 0.09999999999999998, 0.1428571428571429, 0.17
24137931034483, 0.36363636363636365, 0.15625, 0.1428571428571429, 0.3000000000000000
4, 0.09999999999999998, 0.1785714285714286, 0.1428571428571429, 0.09999999999999998,
0.2272727272727273, 0.1875, 0.1428571428571429, 0.1428571428571429, 0.08333333333333
337, 1.0, 0.0, 0.09090909090909094, 0.09999999999999998, 0.05882352941176472, 0.0666
6666666666665, 0.11111111111111116, 0.25, 0.21052631578947367, 0.0, 0.0, 0.0, 0.0769
2307692307687, 0.11538461538461542, 0.09999999999999998, 0.1428571428571429, 0.11111
111111111116, 0.125, 0.11111111111111116, 0.125, 0.0, 0.13793103448275867, 0.0909090
9090909094, 0.09375, 0.11428571428571432, 0.1428571428571429, 0.09999999999999998, 0
.1071428571428571, 0.08571428571428574, 0.11111111111111116, 0.045454545454545414, 0
.0625, 0.0, 0.23076923076923073, 0.3076923076923077, 0.0, 0.5384615384615384, 0.0, 0
.46153846153846156, 0.4117647058823529, 0.4666666666666667, 0.5, 0.23076923076923073
, 0.052631578947368474, 0.3846153846153846, 0.15384615384615385, 0.1333333333333333,
0.07692307692307687, 0.11538461538461542, 0.23076923076923073, 0.0, 0.30769230769230
77, 0.125, 0.2592592592592593, 0.3076923076923077, 0.0, 0.1724137931034483, 0.307692
3076923077, 0.15625, 0.1428571428571429, 0.23076923076923073, 0.15384615384615385, 0
.2142857142857143, 0.17142857142857137, 0.15384615384615385, 0.18181818181818177, 0.
1875, 0.0, 0.25, 0.41666666666666663, 0.125, 0.09090909090909094, 0.1818181818181817
7, 0.09999999999999998, 0.17647058823529416, 0.1333333333333333, 0.11111111111111116
, 0.0, 0.052631578947368474, 0.19999999999999996, 0.09999999999999998, 0.13333333333
33333, 0.15384615384615385, 0.11538461538461542, 0.0, 0.25, 0.11111111111111116, 0.1
875, 0.07407407407407407, 0.0, 0.0714285714285714, 0.10344827586206895, 0.2727272727
272727, 0.09375, 0.11428571428571432, 0.125, 0.19999999999999996, 0.1071428571428571
, 0.11428571428571432, 0.11111111111111116, 0.13636363636363635, 0.125, 0.0666666666
6666665, 0.1333333333333333, 0.06666666666666665, 0.19999999999999996, 0.13333333333
33333, 0.1333333333333333, 0.1333333333333333, 0.05882352941176472, 0.0, 0.166666666
66666663, 0.1333333333333333, 0.1578947368421053, 0.06666666666666665, 0.13333333333
33333, 0.06666666666666665, 0.1333333333333333, 0.3076923076923077, 0.06666666666666
665, 0.06666666666666665, 0.1333333333333333, 0.125, 0.11111111111111116, 0.13333333
33333333, 0.1333333333333333, 0.27586206896551724, 0.1333333333333333, 0.28125, 0.28
57142857142857, 0.1333333333333333, 0.1333333333333333, 0.2142857142857143, 0.228571
42857142854, 0.1333333333333333, 0.2272727272727273, 0.125, 0.0714285714285714, 0.07
14285714285714, 0.1428571428571429, 0.0, 0.1428571428571429, 0.0714285714285714, 0.2
142857142857143, 0.17647058823529416, 0.1333333333333333, 0.2222222222222222, 0.1428
571428571429, 0.21052631578947367, 0.1428571428571429, 0.1428571428571429, 0.2666666
666666667, 0.1428571428571429, 0.34615384615384615, 0.1428571428571429, 0.0, 0.21428
57142857143, 0.125, 0.11111111111111116, 0.0714285714285714, 0.3571428571428571, 0.2
068965517241379, 0.0714285714285714, 0.1875, 0.17142857142857137, 0.1428571428571429
, 0.0714285714285714, 0.25, 0.19999999999999996, 0.0714285714285714, 0.3181818181818
1823, 0.125, 0.040000000000000036, 0.07999999999999996, 0.16000000000000003, 0.07999
999999999996, 0.19999999999999996, 0.12, 0.19999999999999996, 0.16000000000000003, 0
.19999999999999996, 0.24, 0.16000000000000003, 0.24, 0.12, 0.16000000000000003, 0.36
, 0.24, 0.1923076923076923, 0.24, 0.040000000000000036, 0.16000000000000003, 0.12, 0
```

.14814814814814814, 0.12, 0.28, 0.1724137931034483, 0.07999999999999996, 0.21875, 0.
2571428571428571, 0.12, 0.12, 0.25, 0.22857142857142854, 0.07999999999999996, 0.36,
0.24, 0.0, 0.07692307692307687, 0.07692307692307687, 0.15384615384615385, 0.23076923
076923073, 0.0, 0.3076923076923077, 0.2941176470588235, 0.1333333333333333, 0.111111
11111111116, 0.23076923076923073, 0.21052631578947367, 0.15384615384615385, 0.076923
07692307687, 0.1333333333333333, 0.07692307692307687, 0.15384615384615385, 0.1538461
5384615385, 0.0, 0.23076923076923073, 0.1875, 0.33333333333333337, 0.307692307692307
7, 0.0714285714285714, 0.13793103448275867, 0.15384615384615385, 0.125, 0.1428571428
571429, 0.23076923076923073, 0.07692307692307687, 0.1785714285714286, 0.171428571428
57137, 0.15384615384615385, 0.2727272727272727, 0.0625, 0.05882352941176472, 0.11764
705882352944, 0.17647058823529416, 0.11764705882352944, 0.17647058823529416, 0.05882
352941176472, 0.17647058823529416, 0.05882352941176472, 0.11764705882352944, 0.16666
666666666663, 0.11764705882352944, 0.10526315789473684, 0.05882352941176472, 0.11764
705882352944, 0.11764705882352944, 0.11764705882352944, 0.34615384615384615, 0.17647
058823529416, 0.05882352941176472, 0.2941176470588235, 0.05882352941176472, 0.185185
18518518523, 0.3529411764705882, 0.17647058823529416, 0.27586206896551724, 0.0588235
2941176472, 0.25, 0.3142857142857143, 0.11764705882352944, 0.17647058823529416, 0.17
85714285714286, 0.22857142857142854, 0.05882352941176472, 0.13636363636363635, 0.117
64705882352944, 0.0, 0.09999999999999998, 0.25, 0.09999999999999998, 0.6363636363636
364, 0.36363636363636365, 1.0, 0.5294117647058824, 0.6, 0.4444444444444444, 0.300000
00000000004, 0.10526315789473684, 0.09999999999999998, 0.09999999999999998, 0.133333
3333333333, 0.07692307692307687, 0.15384615384615385, 0.4, 0.19999999999999996, 0.19
999999999999996, 0.1875, 0.2962962962962963, 0.30000000000000004, 0.2142857142857143
, 0.1724137931034483, 0.18181818181818177, 0.15625, 0.1428571428571429, 0.3000000000
0000004, 0.0, 0.2142857142857143, 0.17142857142857137, 0.4, 0.18181818181818177, 0.1
875, 0.0, 0.2222222222222222, 0.08333333333333337, 0.11111111111111116, 0.2727272727
272727, 0.09090909090909094, 0.19999999999999996, 0.5294117647058824, 0.466666666666
6667, 0.2777777777777778, 0.11111111111111116, 0.10526315789473684, 0.19999999999999
996, 0.0, 0.33333333333333337, 0.3846153846153846, 0.1923076923076923, 0.0, 0.111111
11111111116, 1.0, 0.125, 0.18518518518518523, 0.4444444444444444, 0.2142857142857143
, 0.13793103448275867, 0.2727272727272727, 0.125, 0.2571428571428571, 0.444444444444
4444, 0.19999999999999996, 0.2142857142857143, 0.17142857142857137, 0.0, 0.409090909
09090906, 0.125, 0.1428571428571429, 0.2857142857142857, 0.16666666666666663, 0.1428
571428571429, 0.18181818181818177, 0.18181818181818177, 0.30000000000000004, 0.29411
76470588235, 0.4666666666666667, 0.2222222222222222, 0.125, 0.1578947368421053, 0.19
999999999999996, 0.0, 0.2666666666666667, 0.3846153846153846, 0.15384615384615385, 0
.09999999999999998, 0.0, 0.4444444444444444, 0.125, 0.11111111111111116, 0.0, 0.1428
571428571429, 0.13793103448275867, 0.18181818181818177, 0.125, 0.11428571428571432,
1.0, 0.19999999999999996, 0.1071428571428571, 0.08571428571428574, 0.111111111111111
16, 0.2272727272727273, 0.1875, 0.0, 0.2857142857142857, 0.25, 0.0, 0.09090909090909
094, 0.18181818181818177, 0.0, 0.11764705882352944, 0.1333333333333333, 0.1111111111
1111116, 0.125, 0.10526315789473684, 0.09999999999999998, 0.0, 0.1333333333333333, 0
.23076923076923073, 0.11538461538461542, 0.19999999999999996, 0.1428571428571429, 0.
2222222222222222, 0.125, 0.07407407407407407, 0.125, 0.0714285714285714, 0.103448275
86206895, 0.18181818181818177, 0.09375, 0.11428571428571432, 0.1428571428571429, 0.3
0000000000000004, 0.0714285714285714, 0.08571428571428574, 0.0, 0.09090909090909094,
0.3125, 0.09090909090909094, 0.18181818181818177, 0.08333333333333337, 0.09090909090
909094, 0.09090909090909094, 0.0, 0.18181818181818177, 0.23529411764705888, 0.333333
33333333337, 0.2222222222222222, 0.09090909090909094, 0.052631578947368474, 0.181818
18181818177, 0.0, 0.19999999999999996, 0.23076923076923073, 0.1923076923076923, 0.18
181818181818177, 0.09090909090909094, 0.36363636363636365, 0.125, 0.1481481481481481
4, 0.09090909090909094, 0.1428571428571429, 0.10344827586206895, 0.09090909090909094
, 0.09375, 0.1428571428571429, 0.2727272727272727, 0.18181818181818177, 0.1428571428
571429, 0.11428571428571432, 0.09090909090909094, 0.2272727272727273, 0.1875, 0.0, 0
.0714285714285714, 0.2142857142857143, 0.1428571428571429, 0.5, 0.0714285714285714,
0.5714285714285714, 0.7058823529411764, 0.6, 0.5, 0.2857142857142857, 0.210526315789

47367, 0.2142857142857143, 0.1428571428571429, 0.1333333333333333, 0.142857142857142
9, 0.23076923076923073, 0.1428571428571429, 0.0, 0.3571428571428571, 0.1875, 0.51851
85185185186, 0.2857142857142857, 0.0714285714285714, 0.2068965517241379, 0.214285714
2857143, 0.1875, 0.17142857142857137, 0.2142857142857143, 0.1428571428571429, 0.2857
142857142857, 0.22857142857142854, 0.1428571428571429, 0.2727272727272727, 0.0625, 0
.04000000000000036, 0.07999999999999996, 0.24, 0.12, 0.31999999999999995, 0.0799999
9999999996, 0.36, 0.52, 0.4, 0.4, 0.16000000000000003, 0.16000000000000003, 0.160000
00000000003, 0.12, 0.16000000000000003, 0.16000000000000003, 0.23076923076923073, 0.
16000000000000003, 0.040000000000000036, 0.24, 0.12, 0.11111111111111116, 0.24, 0.16
000000000000003, 0.1724137931034483, 0.16000000000000003, 0.1875, 0.2857142857142857
, 0.16000000000000003, 0.12, 0.2142857142857143, 0.19999999999999996, 0.12, 0.199999
99999999996, 0.24, 0.09999999999999998, 0.4, 0.16666666666666663, 0.0, 0.18181818181
818177, 0.09090909090909094, 0.09999999999999998, 0.23529411764705888, 0.26666666666
66667, 0.2222222222222222, 0.09999999999999998, 0.3157894736842105, 1.0, 0.6, 0.4, 0
.23076923076923073, 0.07692307692307687, 0.0, 0.09999999999999998, 0.199999999999999
96, 0.25, 0.11111111111111116, 0.0, 0.1428571428571429, 0.13793103448275867, 0.54545
45454545454, 0.125, 0.11428571428571432, 0.19999999999999996, 0.0, 0.142857142857142
9, 0.11428571428571432, 0.19999999999999996, 0.2272727272727273, 0.125, 0.0769230769
2307687, 0.07692307692307687, 0.15384615384615385, 0.15384615384615385, 0.1538461538
4615385, 0.07692307692307687, 0.15384615384615385, 0.17647058823529416, 0.1333333333
333333, 0.16666666666666663, 0.23076923076923073, 0.631578947368421, 0.5384615384615
384, 0.46153846153846156, 0.2666666666666667, 0.15384615384615385, 0.153846153846153
85, 0.07692307692307687, 0.15384615384615385, 0.15384615384615385, 0.375, 0.14814814
814814814, 0.07692307692307687, 0.1428571428571429, 0.13793103448275867, 0.153846153
84615385, 0.125, 0.1428571428571429, 0.15384615384615385, 0.15384615384615385, 0.178
5714285714286, 0.19999999999999996, 0.15384615384615385, 0.2272727272727273, 0.1875,
0.07692307692307687, 0.15384615384615385, 0.07692307692307687, 0.07692307692307687,
0.07692307692307687, 0.15384615384615385, 0.07692307692307687, 0.3529411764705882, 0
.4, 0.2222222222222222, 0.15384615384615385, 0.21052631578947367, 0.2307692307692307
3, 0.07692307692307687, 0.4666666666666667, 1.0, 0.1923076923076923, 0.1538461538461
5385, 0.15384615384615385, 0.3846153846153846, 0.1875, 0.14814814814814814, 0.153846
15384615385, 0.0714285714285714, 0.13793103448275867, 0.15384615384615385, 0.15625,
0.1428571428571429, 0.3846153846153846, 0.23076923076923073, 0.1785714285714286, 0.1
428571428571429, 0.07692307692307687, 0.31818181818181823, 0.25]
Best matches for Cassandra_Postgres:
Jaccard Similarity: 0.05714285714285714
Levenshtein Similarity: [1.0, 0.125, 0.09090909090909094, 0.0, 0.19999999999999996,
0.0, 0.21052631578947367, 0.2777777777777778, 0.2777777777777778, 0.1111111111111111
6, 0.1428571428571429, 0.08333333333333337, 0.16666666666666663, 0.18181818181818177
, 0.09090909090909094, 0.2666666666666667, 0.125, 0.6, 0.4, 0.09090909090909094, 0.1
9999999999999996, 0.0, 0.19999999999999996, 0.26315789473684215, 0.33333333333333337
, 0.33333333333333337, 0.09999999999999998, 0.1428571428571429, 0.16666666666666663,
0.08333333333333337, 0.36363636363636365, 0.0, 0.19999999999999996, 0.19999999999999
996, 0.0, 0.1428571428571429, 0.6363636363636364, 0.0, 0.09999999999999998, 0.125, 0
.10526315789473684, 0.05555555555555558, 0.05555555555555558, 0.11111111111111116, 0
.1428571428571429, 0.08333333333333337, 0.0, 0.0, 0.0, 0.0, 0.0, 0.15384615384615385
, 0.23076923076923073, 0.0, 0.3076923076923077, 0.07692307692307687, 0.1538461538461
5385, 0.26315789473684215, 0.2222222222222222, 0.2222222222222222, 0.230769230769230
73, 0.1428571428571429, 0.07692307692307687, 0.15384615384615385, 0.0769230769230768
7, 0.23076923076923073, 0.06666666666666665, 0.07692307692307687, 0.0, 0.25, 0.09090
909090909094, 0.125, 0.0, 1.0, 0.10526315789473684, 0.11111111111111116, 0.111111111
11111116, 0.4444444444444444, 0.1428571428571429, 0.16666666666666663, 0.08333333333
333337, 0.09090909090909094, 0.09090909090909094, 0.06666666666666665, 0.0, 0.133333
3333333333, 0.1333333333333333, 0.06666666666666665, 0.1333333333333333, 0.133333333
3333333, 0.33333333333333337, 0.10526315789473684, 0.16666666666666663, 0.2222222222
222222, 0.1333333333333333, 0.06666666666666665, 0.1333333333333333, 0.1333333333333

```
333, 0.06666666666666665, 0.1333333333333333, 0.1333333333333333, 0.0666666666666666
5, 0.2857142857142857, 0.0714285714285714, 0.0, 0.1428571428571429, 0.14285714285714
29, 0.0714285714285714, 0.21052631578947367, 0.2777777777777778, 0.2222222222222222,
0.1428571428571429, 0.0714285714285714, 0.1428571428571429, 0.1428571428571429, 0.21
42857142857143, 0.0714285714285714, 0.19999999999999996, 0.0714285714285714, 0.19999
999999999996, 0.07999999999999996, 0.07999999999999996, 0.16000000000000003, 0.12, 0
.07999999999999996, 0.24, 0.28, 0.24, 0.19999999999999996, 0.16000000000000003, 0.16
000000000000003, 0.16000000000000003, 0.16000000000000003, 0.07999999999999996, 0.19
999999999999996, 0.07999999999999996, 0.15384615384615385, 0.07692307692307687, 0.07
692307692307687, 0.15384615384615385, 0.07692307692307687, 0.07692307692307687, 0.10
526315789473684, 0.16666666666666663, 0.16666666666666663, 0.07692307692307687, 0.0,
0.15384615384615385, 0.15384615384615385, 0.23076923076923073, 0.07692307692307687,
0.19999999999999996, 0.23076923076923073, 0.11764705882352944, 0.11764705882352944,
0.11764705882352944, 0.11764705882352944, 0.11764705882352944, 0.11764705882352944,
0.052631578947368474, 0.16666666666666663, 0.16666666666666663, 0.23529411764705888,
0.11764705882352944, 0.17647058823529416, 0.11764705882352944, 0.05882352941176472,
0.11764705882352944, 0.11764705882352944, 0.17647058823529416, 0.4, 0.09999999999999
998, 0.0, 0.30000000000000004, 0.09999999999999998, 0.09999999999999998, 0.210526315
78947367, 0.2222222222222222, 0.2222222222222222, 0.09999999999999998, 0.21428571428
57143, 0.08333333333333337, 0.16666666666666663, 0.09090909090909094, 0.090909090909
09094, 0.19999999999999996, 0.09999999999999998, 0.0, 0.2222222222222222, 0.09090909
090909094, 0.2222222222222222, 0.0, 0.11111111111111116, 0.10526315789473684, 0.3333
3333333333337, 0.33333333333333337, 0.2222222222222222, 0.2142857142857143, 0.333333
33333333337, 0.16666666666666663, 0.36363636363636365, 0.18181818181818177, 0.133333
3333333333, 0.2222222222222222, 0.125, 0.2857142857142857, 0.09090909090909094, 0.12
5, 0.0, 0.125, 0.10526315789473684, 0.2222222222222222, 0.2222222222222222, 0.111111
11111111116, 0.1428571428571429, 0.33333333333333337, 0.08333333333333337, 0.4545454
545454546, 0.0, 0.19999999999999996, 0.125, 0.125, 0.2857142857142857, 0.09090909090
909094, 0.25, 0.09999999999999998, 0.125, 0.21052631578947367, 0.2222222222222222, 0
.11111111111111116, 0.4444444444444444, 0.1428571428571429, 0.16666666666666663, 0.0
8333333333333337, 0.18181818181818177, 0.09090909090909094, 0.1333333333333333, 0.0,
0.09090909090909094, 0.18181818181818177, 0.09090909090909094, 0.36363636363636365,
0.0, 0.09090909090909094, 0.10526315789473684, 0.33333333333333337, 0.16666666666666
663, 0.36363636363636365, 0.1428571428571429, 0.25, 0.16666666666666663, 0.181818181
81818177, 0.0, 0.1333333333333333, 0.09090909090909094, 0.1428571428571429, 0.071428
5714285714, 0.0714285714285714, 0.2142857142857143, 0.0714285714285714, 0.0714285714
285714, 0.21052631578947367, 0.2222222222222222, 0.2222222222222222, 0.1428571428571
429, 0.1428571428571429, 0.1428571428571429, 0.1428571428571429, 0.1428571428571429,
0.2142857142857143, 0.06666666666666665, 0.1428571428571429, 0.12, 0.079999999999999
96, 0.07999999999999996, 0.16000000000000003, 0.12, 0.12, 0.19999999999999996, 0.160
000000000000003, 0.12, 0.16000000000000003, 0.16000000000000003, 0.12, 0.12, 0.12, 0.
19999999999999996, 0.12, 0.12, 0.19999999999999996, 0.4, 0.09090909090909094, 0.4, 0
.19999999999999996, 0.19999999999999996, 0.21052631578947367, 0.2777777777777778, 0.
2777777777777778, 0.09999999999999998, 0.1428571428571429, 0.16666666666666663, 0.16
666666666666663, 0.36363636363636365, 0.09090909090909094, 0.19999999999999996, 0.09
999999999999998, 0.23076923076923073, 0.07692307692307687, 0.15384615384615385, 0.30
76923076923077, 0.23076923076923073, 0.15384615384615385, 0.21052631578947367, 0.166
66666666666663, 0.16666666666666663, 0.07692307692307687, 0.07142857142857142, 0.0769
2307692307687, 0.0, 0.23076923076923073, 0.07692307692307687, 0.19999999999999996, 0
.23076923076923073, 0.23076923076923073, 0.15384615384615385, 0.07692307692307687, 0
.15384615384615385, 0.15384615384615385, 0.15384615384615385, 0.26315789473684215, 0
.4444444444444444, 0.4444444444444444, 0.23076923076923073, 0.1428571428571429, 0.30
76923076923077, 0.15384615384615385, 0.3846153846153846, 0.07692307692307687, 0.0666
6666666666665, 0.15384615384615385]
```

In [24]:
```
# Step 1: Define the Global Schema (G)
```

```
global_schema = {
    'ClientID': str,
    'Name': str,
    'Address': str,
    'ContactInfo': str,
    'ProgramName': str,
    'StartDate': str,
    'EndDate': str,
    'Status': str,
    'PolicyName': str,
    'BillingAmount': float,
    'LastInspectionDate': str
}

# Example global schema (adjust as needed based on your use case)
```

In [25]:
```
# Step 2: LAV Transformation Logic
def lav_transform(df, schema_mapping):
    """
    Transforms the local dataset into the global schema view using a provided mappi
    """
    transformed_data = {}

    for global_col, local_col in schema_mapping.items():
        if local_col in df.columns:
            transformed_data[global_col] = df[local_col]
        else:
            # Set as None if the local column doesn't exist
            transformed_data[global_col] = None

    return pd.DataFrame([transformed_data])


# Define schema mappings for LAV
schema_mapping_cassandra = {
    'ClientID': 'ClientID', 'Name': 'ClientName', 'Address': 'Address', 'ContactInf
    'ProgramName': 'PolicyName', 'StartDate': 'StartDate', 'EndDate': 'EndDate', 'S
    'PolicyName': 'PolicyName', 'BillingAmount': 'BaseFee', 'LastInspectionDate': '
}

# Apply LAV transformation to Cassandra
lav_cassandra = lav_transform(merged_cassandra, schema_mapping_cassandra)
lav_cassandra.head()  # Display the transformed data
```

Out[25]:

	ClientID	Name	Address	ContactInfo	ProgramName	StartDate	
0	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 Customer_41 1 Customer_42 ...	0 Rua 4, Faro 1 Rua 4, Faro 2 ...	0 35191000004 1 35191000004 2 3519...	0 Policy 3 1 Policy 1 2 Policy 2 3...	0 2024-01-26 1 2024-01-26 2 2024-01-2...	20 20 20

In [26]:
```
def reset_index_with_check(df):
    """
```

```
    Reset the index of the dataframe and ensure the index is unique.
    """
    df_reset = df.reset_index(
        drop=True)  # Reset the index and drop the old one
    if not df_reset.index.is_unique:
        print("Warning: The index is not unique in this dataframe!")
    return df_reset


# Example schema mappings for LAV transformations (adjust these to match your data)
schema_mapping_mongo = {
    '_id': 'ClientID',  # Renaming '_id' to 'ClientID' since '_id' is the actual co
    'Name': 'Name',
    'Address': 'Address',
    'ContactInfo': 'ContactInfo',
    'ProgramName': 'ProgramName',
    'StartDate': 'StartDate',
    'EndDate': 'EndDate',
    'Status': 'Status',
    'PolicyName': 'PolicyName',
    'BillingAmount': 'ProjectedSavings',
    'LastInspectionDate': 'PublicationDate'
}

schema_mapping_postgres = {
    'ClientID': 'ClientID',
    'Name': 'Name',
    'Address': 'FullAddress',
    'ContactInfo': 'MobileInfo',
    'ProgramName': 'WasteType',
    'StartDate': 'LastCollectionDate',
    'EndDate': 'NextCollectionDate',
    'Status': 'Status',
    'PolicyName': 'WasteType',
    'BillingAmount': 'TotalAmount',
    'LastInspectionDate': 'BillingDate'
}

# Updated schema mapping for Web dataset based on the column names you provided
schema_mapping_web = {
    'ReportID': 'ClientID',  # Mapping ReportID to ClientID
    'Name': 'SensorType',
    'Address': 'Location',
    'ContactInfo': 'ReportURL',
    'ProgramName': 'Temperature',
    'StartDate': 'Time',
    'EndDate': 'Time',
    'Status': 'Validation',
    'PolicyName': 'pH',
    'BillingAmount': 'Turbidity',
    'LastInspectionDate': 'Time'
}
```

In [27]:
```
# Apply LAV transformation for Mongo, Postgres, and Web datasets
lav_mongo = lav_transform(merged_mongo, schema_mapping_mongo)
lav_postgres = lav_transform(merged_postgres, schema_mapping_postgres)
```

```
lav_web = lav_transform(merged_web, schema_mapping_web)
```

In [28]:
```
lav_mongo.columns
```

Out[28]:
```
Index(['_id', 'Name', 'Address', 'ContactInfo', 'ProgramName', 'StartDate',
       'EndDate', 'Status', 'PolicyName', 'BillingAmount',
       'LastInspectionDate'],
      dtype='object')
```

In [29]:
```
# Reset the index of the dataframe and ensure the index is unique.

# Step 3: GAV Transformation Logic
def reset_index_with_check(df):
    """
    Reset the index of the dataframe and ensure the index is unique.
    """
    df_reset = df.reset_index(
        drop=True)  # Reset the index and drop the old one
    if not df_reset.index.is_unique:
        print("Warning: The index is not unique in this dataframe!")
    return df_reset

# Step 3: GAV Transformation Logic


def gav_transform(local_dfs):
    """
    Generates a global schema view by combining data from all local sources.
    The local data sources should be merged into a single view.
    """
    # Reset index for each DataFrame to avoid invalid index errors during concatena
    local_dfs_reset = [reset_index_with_check(df) for df in local_dfs]

    # Concatenate the dataframes with ignore_index=True to ensure a unique index fo
    combined_data = pd.concat(local_dfs_reset, ignore_index=True, sort=False)

    # Fill missing data with NaN (the standard placeholder in pandas)
    return combined_data.fillna(np.nan)


# Example schema mappings for LAV transformations (adjust these to match your data)
schema_mapping_mongo = {
    '_id': 'ClientID',  # Renaming '_id' to 'ClientID' since '_id' is the actual co
    'Name': 'Name',
    'Address': 'Address',
    'ContactInfo': 'ContactInfo',
    'ProgramName': 'ProgramName',
    'StartDate': 'StartDate',
    'EndDate': 'EndDate',
    'Status': 'Status',
    'PolicyName': 'PolicyName',
    # Renaming 'BillingAmount' to 'ProjectedSavings'
    'BillingAmount': 'ProjectedSavings',
    # Renaming 'LastInspectionDate' to 'PublicationDate'
    'LastInspectionDate': 'PublicationDate'
}
```

```
schema_mapping_postgres = {
    'ClientID': 'ClientID',
    'Name': 'Name',
    'Address': 'FullAddress',  # Renaming 'Address' to 'FullAddress'
    'ContactInfo': 'MobileInfo',  # Renaming 'ContactInfo' to 'MobileInfo'
    'ProgramName': 'WasteType',
    'StartDate': 'LastCollectionDate',  # Renaming 'StartDate' to 'LastCollectionDa
    'EndDate': 'NextCollectionDate',  # Renaming 'EndDate' to 'NextCollectionDate'
    'Status': 'Status',
    'PolicyName': 'WasteType',  # Keeping 'WasteType' as 'PolicyName'
    'BillingAmount': 'TotalAmount',  # Renaming 'BillingAmount' to 'TotalAmount'
    # Renaming 'LastInspectionDate' to 'BillingDate'
    'LastInspectionDate': 'BillingDate'
}

# Updated schema mapping for Web dataset based on the column names you provided
schema_mapping_web = {
    'ReportID': 'ClientID',  # Mapping ReportID to ClientID
    'Name': 'SensorType',
    'Address': 'Location',
    'ContactInfo': 'ReportURL',
    'ProgramName': 'Temperature',  # Mapping 'ProgramName' to 'Temperature'
    'StartDate': 'Time',
    'EndDate': 'Time',
    'Status': 'Validation',
    'PolicyName': 'pH',  # Mapping 'PolicyName' to 'pH'
    'BillingAmount': 'Turbidity',  # Mapping 'BillingAmount' to 'Turbidity'
    'LastInspectionDate': 'Time'  # Mapping 'LastInspectionDate' to 'Time'
}

# Assuming the merged datasets have already been loaded as DataFrames
# Apply LAV transformation for Mongo, Postgres, and Web datasets
lav_mongo = lav_transform(merged_mongo, schema_mapping_mongo)
lav_postgres = lav_transform(merged_postgres, schema_mapping_postgres)
lav_web = lav_transform(merged_web, schema_mapping_web)


# Ensure lav_cassandra is defined and has a unique index
# Apply the same transformation logic to lav_cassandra as done with other datasets:
# Example (adjust schema_mapping for cassandra):
lav_cassandra = lav_transform(
    merged_cassandra, schema_mapping_cassandra)  # Assuming you have this

# Step 3: GAV Transformation Logic
local_dfs = [lav_cassandra, lav_mongo, lav_postgres,
             lav_web]  # Ensure lav_cassandra is defined too
gav_view = gav_transform(local_dfs)

# Display the GAV view's first few rows
print(gav_view.head())
```

```
                                                ClientID  \
0  0     ea614ade-9cee-43ba-bb90-319f7079f8dc
1  ...
1                                                    NaN
2  0     1
1     1
2     1
3     1
4     1
5     ...
3                                                    NaN

                                                   Name  \
0  0       Customer_4
1       Customer_4
2       ...
1  0      Customer_1
1     Customer_2
2     Cus...
2  0     Customer_1
1     Customer_1
2     Custom...
3  0       Turbidity
1       Turbidity
2       Tu...

                                                Address  \
0  0       Rua 4, Faro
1       Rua 4, Faro
2     ...
1  0      rua 1, Faro
1     rua 2, Faro
2     r...
2  0     rua 1, Faro
1     rua 1, Faro
2     rua ...
3  0     Reservoir_5
1     Reservoir_3
2     Rese...

                                              ContactInfo  \
0  0     35191000004
1     35191000004
2     3519...
1  0     35191000001
1     35191000002
2     3519...
2  0     35191000001
1     35191000001
2     3519...
3  0      report_1.pdf
1     report_2.pdf
2     ...

                                              ProgramName  \
0  0     Policy 3
```

```
1     Policy 1
2     Policy 2
3...
1  0      Program_1
1       Program_2
2       Progr...
2  0         Organic
1         Organic
2         Org...
3  0     26.86
1     10.25
2     10.25
3     34.9...

                                                StartDate  \
0  0    2024-01-26
1    2024-01-26
2    2024-01-2...
1  0     2023-01-04
1     2023-01-07
2     2023-0...
2  0    2025-01-24
1    2025-01-24
2    2025-01-2...
3  0    2024-01-15
1    2024-08-27
2    2024-08-2...

                                                  EndDate  \
0  0    2026-01-25
1    2026-01-25
2    2026-01-2...
1  0     2025-01-31
1     2025-03-02
2     2025-0...
2  0    2025-01-25
1    2025-01-25
2    2025-01-2...
3  0    2024-01-15
1    2024-08-27
2    2024-08-2...

                                                   Status  \
0  0     Terminated
1     Terminated
2     Termin...
1                                                    NaN
2                                                    NaN
3  0     0
1     0
2     0
3     1
4     1
5     ...

                                                PolicyName  \
```

```
0  0     Policy 3
1     Policy 1
2     Policy 2
3...
1  0      Policy_1
1      Policy_2
2      Policy_...
2  0        Organic
1        Organic
2        Org...
3  0     7.42
1     8.40
2     8.40
3     7.70
4 ...

                                      BillingAmount  \
0  0     100.0
1     100.0
2     100.0
3     100....
1  0      1.5
1      3.0
2      4.5
3      6.0
4 ...
2  0     5.0
1     5.0
2     5.0
3     5.0
4     ...
3  0     5.30
1     5.68
2     5.68
3     2.67
4 ...

                                  LastInspectionDate  _id  ReportID
0  0    2025-01-25 17:03:24.432
1    2025-01-25 1...  NaN       NaN
1  0    2022-06-01
1    2022-11-01
2           Na...  NaN       NaN
2  0    2025-01-24
1    2025-01-24
2    2025-01-2...  NaN       NaN
3  0    2024-01-15
1    2024-08-27
2    2024-08-2...  NaN       NaN
C:\Users\iamro\AppData\Local\Temp\ipykernel_21252\3554554553.py:29: FutureWarning: D
owncasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will cha
nge in a future version. Call result.infer_objects(copy=False) instead. To opt-in to
the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`
  return combined_data.fillna(np.nan)
```

In [30]:
```
def glav_transform(global_df, local_dfs):
    """
    Transforms data based on both global and local schemas.
    Global schema is mapped over local data.
    """
    # Merge the global schema with local data
    merged_data = pd.concat([global_df] + local_dfs,
                            ignore_index=True, sort=False)

    # Optional: If you don't want to fill missing values, just return the merged da
    return merged_data



# Apply GLAV transformation by merging global schema with local data
# Combine global and local data
glav_view = glav_transform(gav_view, local_dfs)

# Display the GLAV view's first few rows
print(glav_view.head())
```

```
                                                ClientID  \
0  0     ea614ade-9cee-43ba-bb90-319f7079f8dc
1  ...
1                                                   NaN
2  0     1
1     1
2     1
3     1
4     1
5     ...
3                                                   NaN
4  0     ea614ade-9cee-43ba-bb90-319f7079f8dc
1  ...

                                                   Name  \
0  0       Customer_4
1       Customer_4
2       ...
1  0      Customer_1
1      Customer_2
2      Cus...
2  0     Customer_1
1     Customer_1
2     Custom...
3  0       Turbidity
1       Turbidity
2       Tu...
4  0       Customer_4
1       Customer_4
2       ...

                                                Address  \
0  0       Rua 4, Faro
1       Rua 4, Faro
2     ...
1  0      rua 1, Faro
1      rua 2, Faro
2      r...
2  0     rua 1, Faro
1     rua 1, Faro
2     rua ...
3  0     Reservoir_5
1     Reservoir_3
2     Rese...
4  0       Rua 4, Faro
1       Rua 4, Faro
2     ...

                                              ContactInfo  \
0  0     35191000004
1     35191000004
2     3519...
1  0     35191000001
1     35191000002
2     3519...
2  0     35191000001
```

```
1     35191000001
2     3519...
3  0      report_1.pdf
1      report_2.pdf
2     ...
4  0     35191000004
1     35191000004
2     3519...

                                          ProgramName  \
0  0     Policy 3
1     Policy 1
2     Policy 2
3...
1  0      Program_1
1      Program_2
2      Progr...
2  0        Organic
1        Organic
2        Org...
3  0     26.86
1     10.25
2     10.25
3     34.9...
4  0     Policy 3
1     Policy 1
2     Policy 2
3...

                                            StartDate  \
0  0    2024-01-26
1    2024-01-26
2    2024-01-2...
1  0     2023-01-04
1     2023-01-07
2     2023-0...
2  0    2025-01-24
1    2025-01-24
2    2025-01-2...
3  0    2024-01-15
1    2024-08-27
2    2024-08-2...
4  0    2024-01-26
1    2024-01-26
2    2024-01-2...

                                              EndDate  \
0  0    2026-01-25
1    2026-01-25
2    2026-01-2...
1  0     2025-01-31
1     2025-03-02
2     2025-0...
2  0    2025-01-25
1    2025-01-25
2    2025-01-2...
```

```
3  0    2024-01-15
1    2024-08-27
2    2024-08-2...
4  0    2026-01-25
1    2026-01-25
2    2026-01-2...

                                                 Status  \
0  0     Terminated
1     Terminated
2     Termin...
1                                                   NaN
2                                                   NaN
3  0     0
1     0
2     0
3     1
4     1
5     ...
4  0     Terminated
1     Terminated
2     Termin...

                                               PolicyName  \
0  0     Policy 3
1     Policy 1
2     Policy 2
3...
1  0      Policy_1
1      Policy_2
2      Policy_...
2  0         Organic
1        Organic
2        Org...
3  0     7.42
1     8.40
2     8.40
3     7.70
4 ...
4  0     Policy 3
1     Policy 1
2     Policy 2
3...

                                              BillingAmount  \
0  0     100.0
1     100.0
2     100.0
3     100....
1  0      1.5
1      3.0
2      4.5
3      6.0
4 ...
2  0     5.0
1     5.0
```

```
2     5.0
3     5.0
4     ...
3  0     5.30
1     5.68
2     5.68
3     2.67
4 ...
4  0     100.0
1     100.0
2     100.0
3     100....

                                LastInspectionDate  _id ReportID
0  0    2025-01-25 17:03:24.432
1    2025-01-25 1...  NaN      NaN
1  0    2022-06-01
1    2022-11-01
2          Na...  NaN      NaN
2  0    2025-01-24
1    2025-01-24
2    2025-01-2...  NaN      NaN
3  0    2024-01-15
1    2024-08-27
2    2024-08-2...  NaN      NaN
4  0    2025-01-25 17:03:24.432
1    2025-01-25 1...  NaN      NaN
```

In [31]:
```
lav_cassandra.columns
```

Out[31]:
```
Index(['ClientID', 'Name', 'Address', 'ContactInfo', 'ProgramName',
       'StartDate', 'EndDate', 'Status', 'PolicyName', 'BillingAmount',
       'LastInspectionDate'],
      dtype='object')
```

In [32]:
```
# Query 1: Retrieve all client information from LAV Cassandra
lav_cassandra[['ClientID', 'Name', 'Address', 'ContactInfo']].head()
```

Out[32]:

	ClientID	Name	Address	ContactInfo
0	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 Customer_4 1 Customer_4 2 ...	0 Rua 4, Faro 1 Rua 4, Faro 2 ...	0 35191000004 1 35191000004 2 3519...

In [33]:
```
# Query 2: Retrieve policies with start and end dates
lav_cassandra[['ClientID', 'PolicyName', 'StartDate', 'EndDate']].head()
```

Out[33]:

	ClientID	PolicyName	StartDate	EndDate
0	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 Policy 3 1 Policy 1 2 Policy 2 3...	0 2024-01-26 1 2024-01-26 2 2024-01-2...	0 2026-01-25 1 2026-01-25 2 2026-01-2...

In [34]:
```
gav_view.columns
```

Out[34]:
```
Index(['ClientID', 'Name', 'Address', 'ContactInfo', 'ProgramName',
       'StartDate', 'EndDate', 'Status', 'PolicyName', 'BillingAmount',
       'LastInspectionDate', '_id', 'ReportID'],
      dtype='object')
```

In [35]:
```
# Query 1: Retrieve all client information from GAV view
gav_view[['ClientID', 'Name', 'Address', 'ContactInfo']].head()
```

Out[35]:

	ClientID	Name	Address	ContactInfo
0	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 Customer_4 1 Customer_4 2 ...	0 Rua 4, Faro 1 Rua 4, Faro 2 ...	0 35191000004 1 35191000004 2 3519...
1	NaN	0 Customer_1 1 Customer_2 2 Cus...	0 rua 1, Faro 1 rua 2, Faro 2 r...	0 35191000001 1 35191000002 2 3519...
2	0 1 1 1 2 1 3 1 4 1 5 ...	0 Customer_1 1 Customer_1 2 Custom...	0 rua 1, Faro 1 rua 1, Faro 2 rua ...	0 35191000001 1 35191000001 2 3519...
3	NaN	0 Turbidity 1 Turbidity 2 Tu...	0 Reservoir_5 1 Reservoir_3 2 Rese...	0 report_1.pdf 1 report_2.pdf 2 ...

In [36]:
```
# Query 2: Retrieve policies with billing amount
gav_view[['PolicyName', 'BillingAmount']].head()
```

Out[36]:

	PolicyName	BillingAmount
0	0 Policy 3 1 Policy 1 2 Policy 2 3...	0 100.0 1 100.0 2 100.0 3 100....
1	0 Policy_1 1 Policy_2 2 Policy_...	0 1.5 1 3.0 2 4.5 3 6.0 4 ...
2	0 Organic 1 Organic 2 Org...	0 5.0 1 5.0 2 5.0 3 5.0 4 ...
3	0 7.42 1 8.40 2 8.40 3 7.70 4 ...	0 5.30 1 5.68 2 5.68 3 2.67 4 ...

In [37]:
```
# Query 3: Retrieve start and end dates for programs
gav_view[['ProgramName', 'StartDate', 'EndDate']].head()
```

Out[37]:

	ProgramName	StartDate	EndDate
0	0 Policy 3 1 Policy 1 2 Policy 2 3...	0 2024-01-26 1 2024-01-26 2 2024-01-2...	0 2026-01-25 1 2026-01-25 2 2026-01-2...
1	0 Program_1 1 Program_2 2 Progr...	0 2023-01-04 1 2023-01-07 2 2023-0...	0 2025-01-31 1 2025-03-02 2 2025-0...
2	0 Organic 1 Organic 2 Org...	0 2025-01-24 1 2025-01-24 2 2025-01-2...	0 2025-01-25 1 2025-01-25 2 2025-01-2...
3	0 26.86 1 10.25 2 10.25 3 34.9...	0 2024-01-15 1 2024-08-27 2 2024-08-2...	0 2024-01-15 1 2024-08-27 2 2024-08-2...

In [38]:
```
# Query 1: Retrieve clients with policy details and program names from GLAV view
glav_view[['ClientID', 'Name', 'PolicyName', 'ProgramName']].head()
```

Out[38]:

	ClientID	Name	PolicyName	ProgramName
0	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 Customer_4 1 Customer_4 2 ...	0 Policy 3 1 Policy 1 2 Policy 2 3...	0 Policy 3 1 Policy 1 2 Policy 2 3...
1	NaN	0 Customer_1 1 Customer_2 2 Cus...	0 Policy_1 1 Policy_2 2 Policy_...	0 Program_1 1 Program_2 2 Progr...
2	0 1 1 1 2 1 3 1 4 1 5 ...	0 Customer_1 1 Customer_1 2 Custom...	0 Organic 1 Organic 2 Org...	0 Organic 1 Organic 2 Org...
3	NaN	0 Turbidity 1 Turbidity 2 Tu...	0 7.42 1 8.40 2 8.40 3 7.70 4 ...	0 26.86 1 10.25 2 10.25 3 34.9...
4	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 Customer_4 1 Customer_4 2 ...	0 Policy 3 1 Policy 1 2 Policy 2 3...	0 Policy 3 1 Policy 1 2 Policy 2 3...

In [39]:
```
glav_view.columns
```

Out[39]:
```
Index(['ClientID', 'Name', 'Address', 'ContactInfo', 'ProgramName',
       'StartDate', 'EndDate', 'Status', 'PolicyName', 'BillingAmount',
       'LastInspectionDate', '_id', 'ReportID'],
      dtype='object')
```

In [40]:
```
# Query 2: Retrieve clients with billing and Policy name
glav_view[['ClientID', 'BillingAmount', 'PolicyName']].head()
```

Out[40]:

	ClientID	BillingAmount	PolicyName
0	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 100.0 1 100.0 2 100.0 3 100....	0 Policy 3 1 Policy 1 2 Policy 2 3...
1	NaN	0 1.5 1 3.0 2 4.5 3 6.0 4 ...	0 Policy_1 1 Policy_2 2 Policy_...
2	0 1 1 1 2 1 3 1 4 1 5 ...	0 5.0 1 5.0 2 5.0 3 5.0 4 ...	0 Organic 1 Organic 2 Org...
3	NaN	0 5.30 1 5.68 2 5.68 3 2.67 4 ...	0 7.42 1 8.40 2 8.40 3 7.70 4 ...
4	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 100.0 1 100.0 2 100.0 3 100....	0 Policy 3 1 Policy 1 2 Policy 2 3...

In [41]:
```
# Query 3: Retrieve clients with last inspection date
glav_view[['ClientID', 'LastInspectionDate']].head()
```

Out[41]:

	ClientID	LastInspectionDate
0	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 2025-01-25 17:03:24.432 1 2025-01-25 1...
1	NaN	0 2022-06-01 1 2022-11-01 2 Na...
2	0 1 1 1 2 1 3 1 4 1 5 ...	0 2025-01-24 1 2025-01-24 2 2025-01-2...
3	NaN	0 2024-01-15 1 2024-08-27 2 2024-08-2...
4	0 ea614ade-9cee-43ba-bb90-319f7079f8dc 1 ...	0 2025-01-25 17:03:24.432 1 2025-01-25 1...