311 Service Request Data Analysis for Kansas City

Executive Summary

This report offers a comprehensive analysis of the 311 service request data from Kansas City, MO, spanning from the inception of the system until March 2021. The dataset contains around 1.56 million records, each representing a distinct service request. The purpose of this analysis is to gain insights into the data's structure, quality, and to identify trends and patterns that can inform operational improvements.

Dataset Overview

The dataset comprises 30 columns and 1,563,215 rows, capturing a wide range of information about each service request, including identifiers, sources, departments, request types, dates, statuses, and geographic details. The Kansas City Dataset is analyzed and visualized using a variety of data points, all of which are included in this report's analytical summary. It is vital to comprehend the structure and quality of the dataset to utilize it efficiently within the business model.

Data Profiling Summary:

Number of Variables: 23
Number of Observations: 1,563,215
Missing Cells: 2,595,087 (7.2%)
Duplicate Rows: 0
Total Memory Size: 274.3 MiB
Average Record Size: 184.0 B

Variable Types:
Numeric: 6
Categorical: 7
Text: 6
DateTime: 3
Data Quality and Cleaning
Key Metrics:

CASE ID: Each service request is uniquely identified by a CASE ID, ensuring no duplication and allowing precise tracking.

SOURCE: The data includes 21 distinct sources of service requests, indicating a diverse range of entry points for citizens' concerns. Major sources include- phone ,web ,and email .
STATUS: The status of service requests is categorized into six distinct values, providing clear visibility into the current state and progress of requests.
DEPARTMENT: The dataset comprises 27 distinct departments responsible for handling service requests, showcasing a wide array of administrative units involved in addressing citizen concerns. Major departments include Public Works, Water Services, Parks and Rec, and Health, among others.

WORK GROUP: There are 146 unique work groups associated with the service requests, highlighting the specific teams or units within departments that manage and resolve the reported issues. This detailed breakdown allows for a granular analysis of the workload distribution and specialization within the various departments.

TYPE: The dataset contains 295 different types of service requests, indicating the diverse nature of issues reported by citizens. This broad categorization enables a detailed understanding of the specific problems or requests submitted by individuals, ranging from infrastructure maintenance to public health concerns.

DETAIL: With 574 distinct details provided for the service requests, the dataset offers a comprehensive view of the specific nature of each reported issue. These detailed descriptions help in understanding the nuances of the problems faced by citizens, facilitating targeted and effective resolution strategies.

CREATION DATE: Each service request is associated with a creation date, ranging from December 29, 2006, to October 28, 2021. This temporal information allows for tracking the timeline of requests and analyzing trends in service request generation over time.

CREATION TIME: The dataset includes creation times for the service requests, ranging from 00:00:00 to 23:59:00. This temporal granularity provides insights into the distribution of request submissions throughout the day, aiding in resource allocation and scheduling of response activities.

EXCEEDED EST TIMEFRAME: A boolean variable indicating whether a service request exceeded the estimated timeframe for resolution. This binary attribute helps in identifying cases where deadlines were not met, highlighting potential areas for process improvement and efficiency enhancement.

CLOSED DATE: The closed date for service requests ranges from January 4, 2007, to February 11, 2022. This temporal information serves as a crucial metric for measuring the turnaround time for request resolution and evaluating the efficiency of service delivery.

DAYS TO CLOSE: The dataset includes the number of days taken to close each service request, with values ranging from -21 to 4525. This quantitative measure provides insights into the timeliness of issue resolution and can be used to assess performance metrics and service level agreements.

Data Quality:

Approximately 7.2% of the dataset is comprised of 2,595,087 missing cells. This indicates a high-quality dataset in terms of completeness, as there is a relatively small amount of missing data. Interestingly, there aren't any duplicate rows, indicating that each observation's uniqueness was preserved during the data collection procedure.

Memory Usage:

The dataset has an average record size of 184.0 B and a total size of 274.3 MiB in memory. These numbers imply that most contemporary computer systems should be able to handle the dataset without the requirement for specialized data processing methods. Six of the variables are numeric, seven are categorical, six are text, three are datetime, and one is boolean. Due to the diversity of data formats, several data pretreatment techniques will be required to ensure that the dataset is prepared appropriately for any processing or analysis.

Possible Consequences:

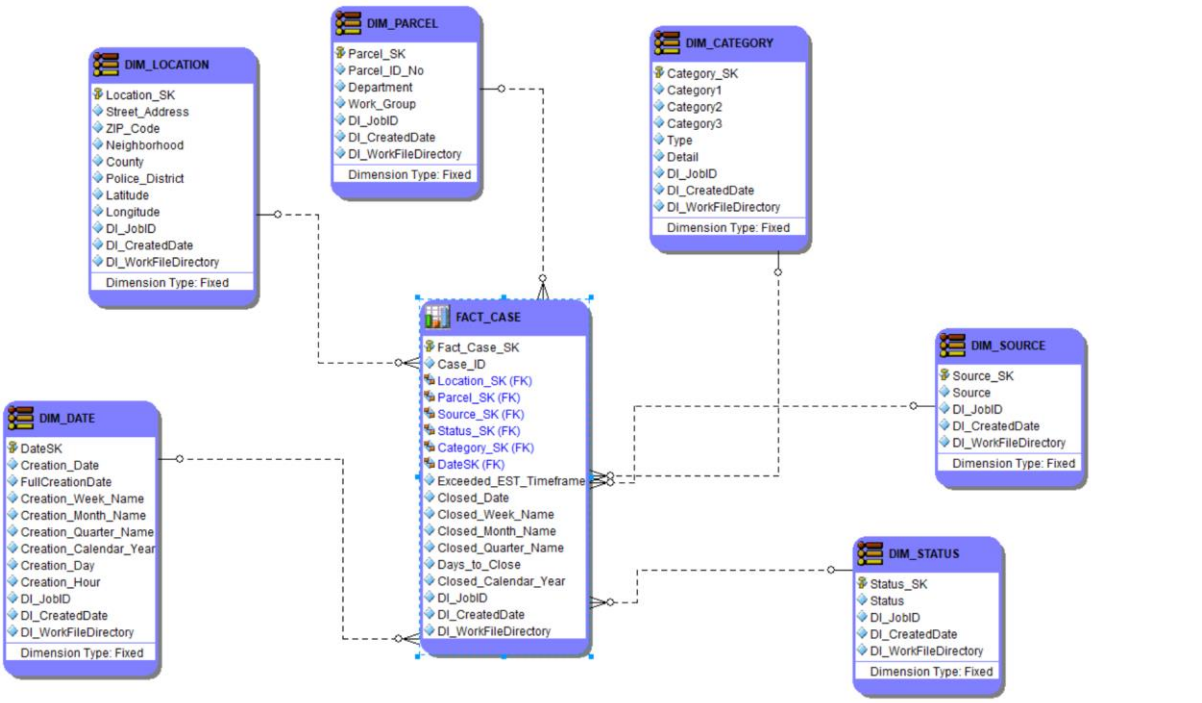
It is unlikely that the proportion of missing data will have a significant effect on the reliability of machine learning models or statistical analysis. Nevertheless, attention should still be paid to dealing with these missing values effectively, either via imputation or exclusion, depending on the analysis's needs. It is advantageous because there are no duplicate rows, which reduces the need for initial data cleansing.

It is advised to investigate the pattern of the missing data before proceeding with the analysis to determine if it is a random omission or if there is a systematic problem that needs to be addressed. If appropriate, consider filling in the missing data with imputation techniques. Furthermore, confirm that each variable's data type corresponds to the format needed for the planned analysis.

Conclusion:

In summary, the dataset appears to be of excellent quality, with a relatively small number of missing values and no duplicate entries. Due to its moderate size, analysis on standard computing systems shouldn't encounter any major difficulties. The dataset is ready for a comprehensive examination or for use in machine learning applications after handling the few missing values correctly.

Data Model:



Dimension and Fact tables:

Dim Source: Details about the origin of each service request.
Dim Status: Standardized statuses of service requests.
Dim Parcel: Geographic parcel information.
Dim Date: Comprehensive date information, with an added SEASON_NAME column.
Dim Location: Geographic details including neighborhood, county, and police district.
Dim Category: Consolidated category information from the original dataset.
Fact Table:

Fact Case: Contains detailed service request information linked to all dimension tables.

Data Cleansing Steps:

Using Intensive Talend ETL/ELT process, given below transformations have been conducted on the dataset to produce meaningful pattern for analytics and vizualisation.
Experimental Transformation-Unpivoting Categories: The categories (CATEGORY1, CATEGORY2, CATEGORY3) were initially unpivoted into a unified category and category description field to simplify and enhance the accuracy of analysis. But later on, found the volume of service requests(case count) were 3 times the initial staged dataset. This model brought easy of querying and accuracy increased tremendously, but came with tradeoff on the overall performance.

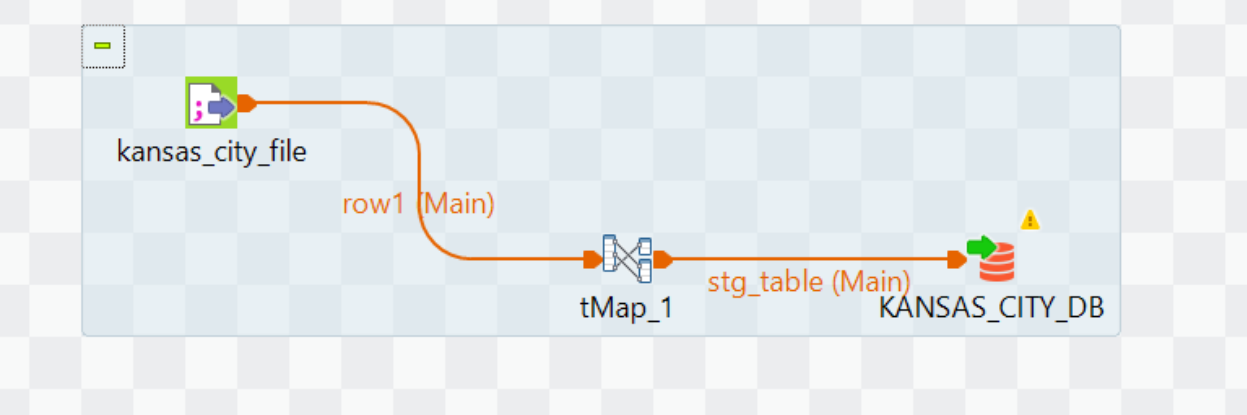
Normalization: All text fields were standardized to a consistent case format to eliminate ambiguities caused by case sensitivity.
Status Categorization: Status values were refined from abbreviations (e.g., canc, dup, assig) to full descriptors (e.g., Canceled, Duplicate, Assigned) for improved readability and analytical utility. When plotting the status to calculate the volume of service request by status , we found the following insights about the dataset behaviour:
for status 'open' - all the closed_date for it is null, as a result days_to_close field is also null
days_to_close have negative values in 29 rows out of which 28 rows are from status 'resolved' and one from 'cancelled'.

Modification of time:
Creation date and time have merged into single column – mergeddate for better query read. ClosedDate and Days_to_close: Since the days_toclose had negative value, the values have been changed to 0 and the corresponding closedate is equated to mergedate(creationdate+creationtime)
Dimensional Modeling
The dataset was structured into a dimensional model using ER Studio, facilitating efficient querying and analysis. The model includes several dimension tables and a central fact table:

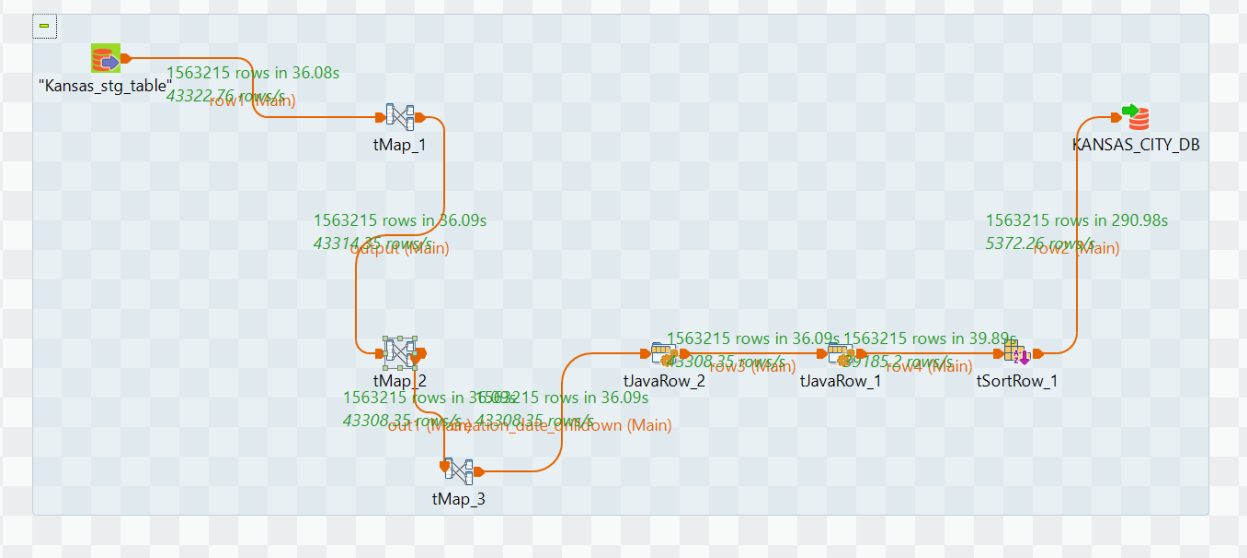
Advanced Analysis

Talend Transformations:

1.Kansas_city_staging table



In the Kansas_staging job, the tsv date file is read using tfiledelimter and using the tmap , it is mapped to staging table, Kansas_stg_table.



After that , Kansas_stg_table is inputed into kansa_city_cleaned table after undergoing several transformations:

1st tmap-convert- trims ,removes special characters and aggregates all the null values in the fields.Formats the date into yyyy-MM-dd type, give meaningful values to status field.

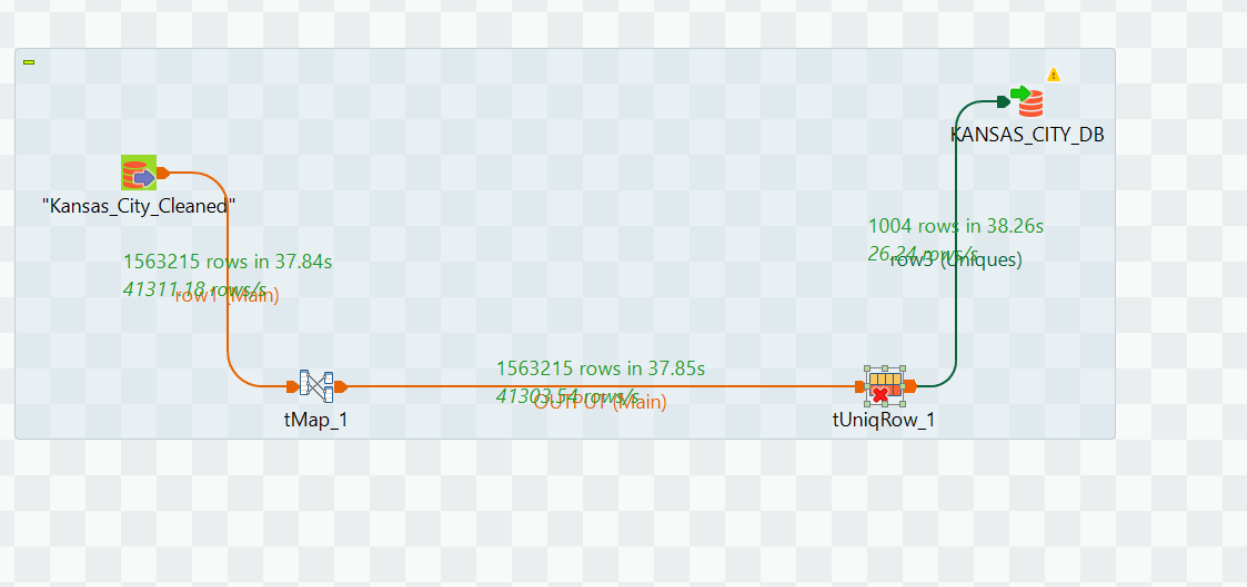
2nd map - combines creation time and creation date to produce mergeddate

3rd map -from merged date , creates week,hours,calender_year,month_name,quarter_name and calender_day

Tjavarow – check if the closed date is not null, then creates the same as done in 3rd tmap.And in the next component , days_to_close is checked if its not null . If so its further checked to identify the value is negative and if true, the closed date is given the merged date

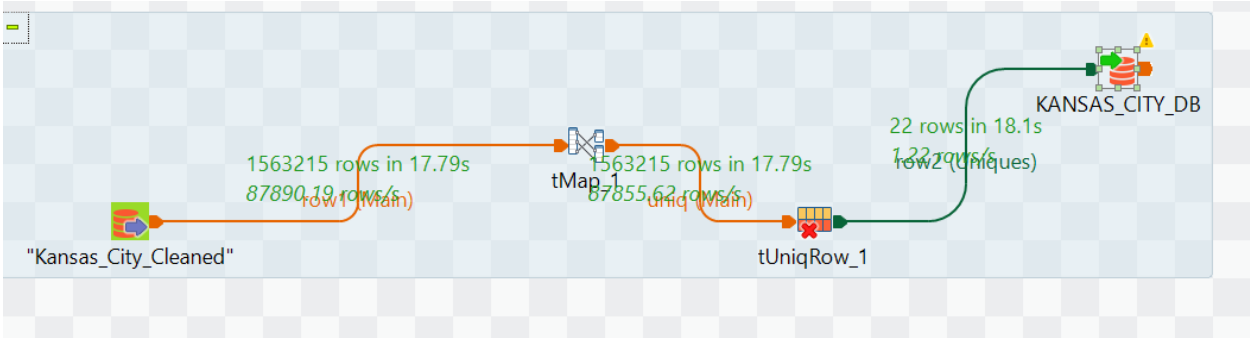
Tsortrow- the rows in dataset is sorted by caseid and creation_date .

2.Dim_Category:



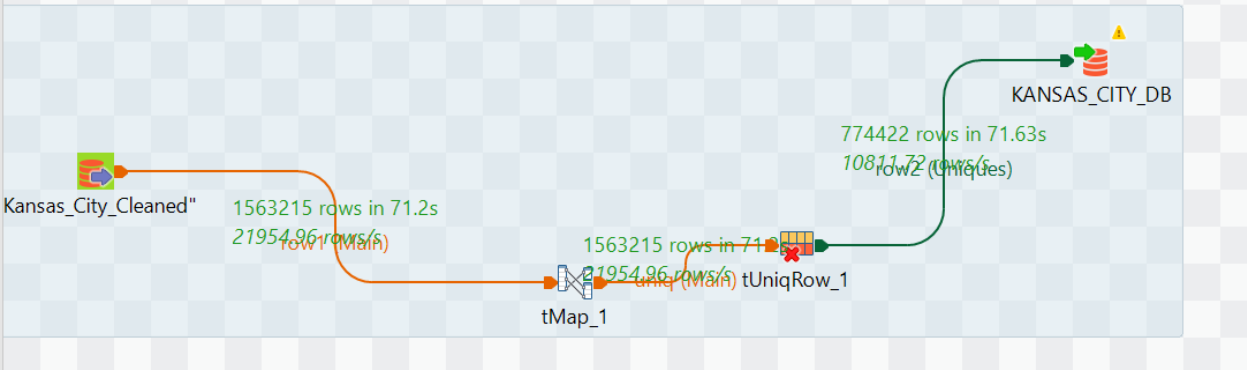
Kansas_city_cleaned table is imported into tmap , mapping the fields: category1, category2, category3,type and detail into Dim_source table .Using tuniq_Row , these field are uniqure inserted into the output table.

3.Dim_Source:



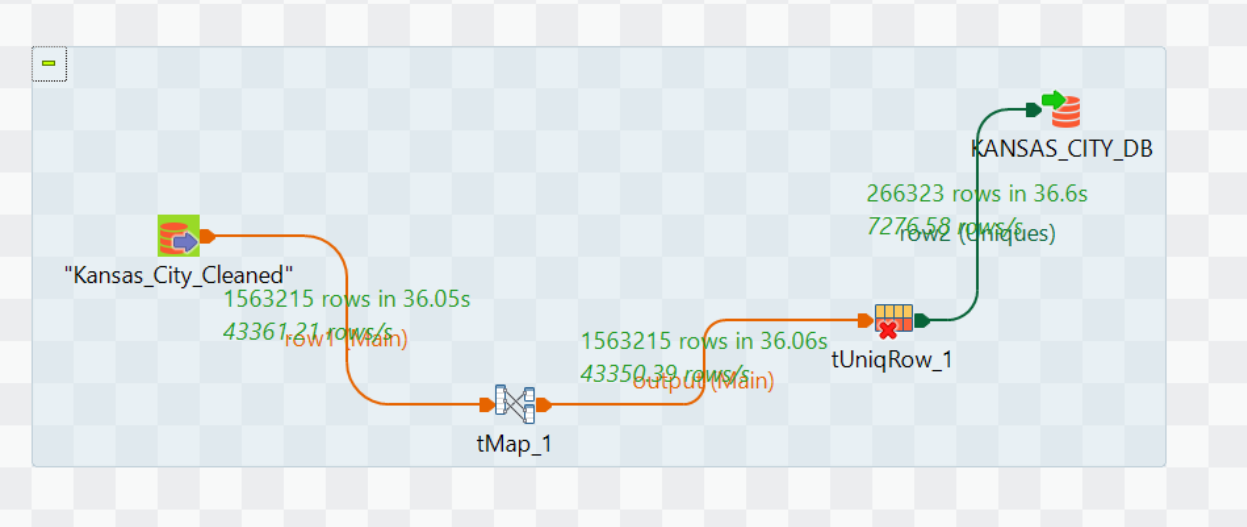
From Kansas_city_cleaned table, the data is derived and directed into tmap and tuniqrow.Later pushed into dim source.

4.Dim_Parcel:



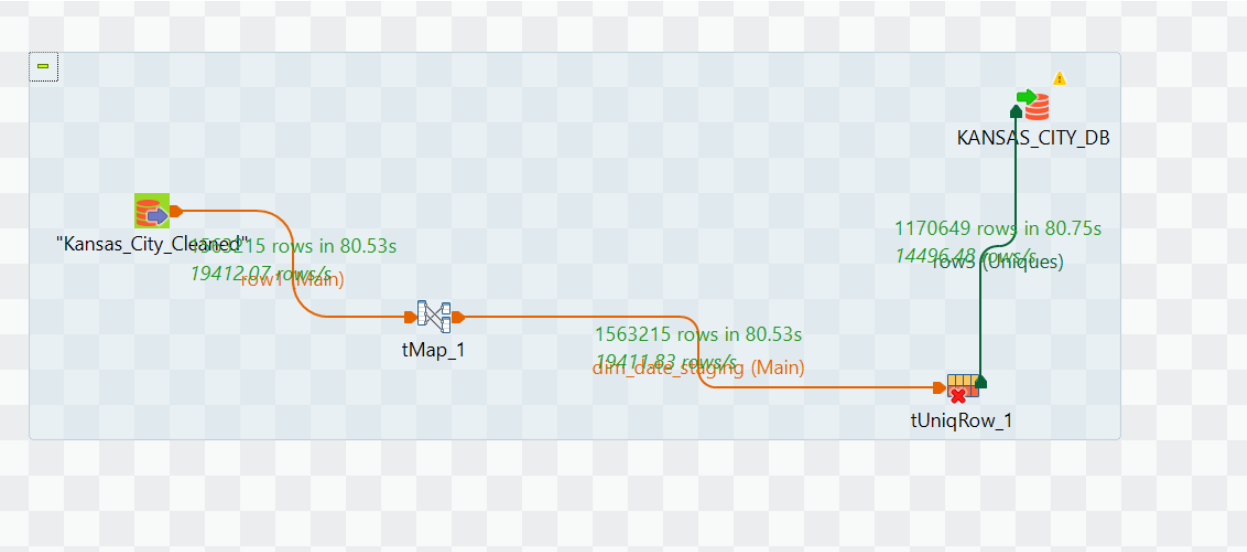
Dim parcel is derived by the same manner as well.

5.Dim_Location:



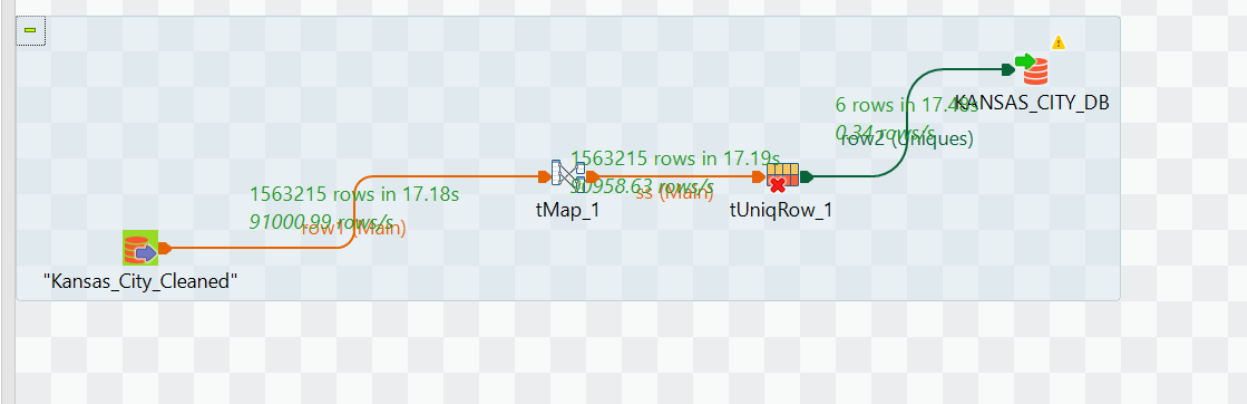
Dim location is derived by the same manner as well.

6.Dim_Date:



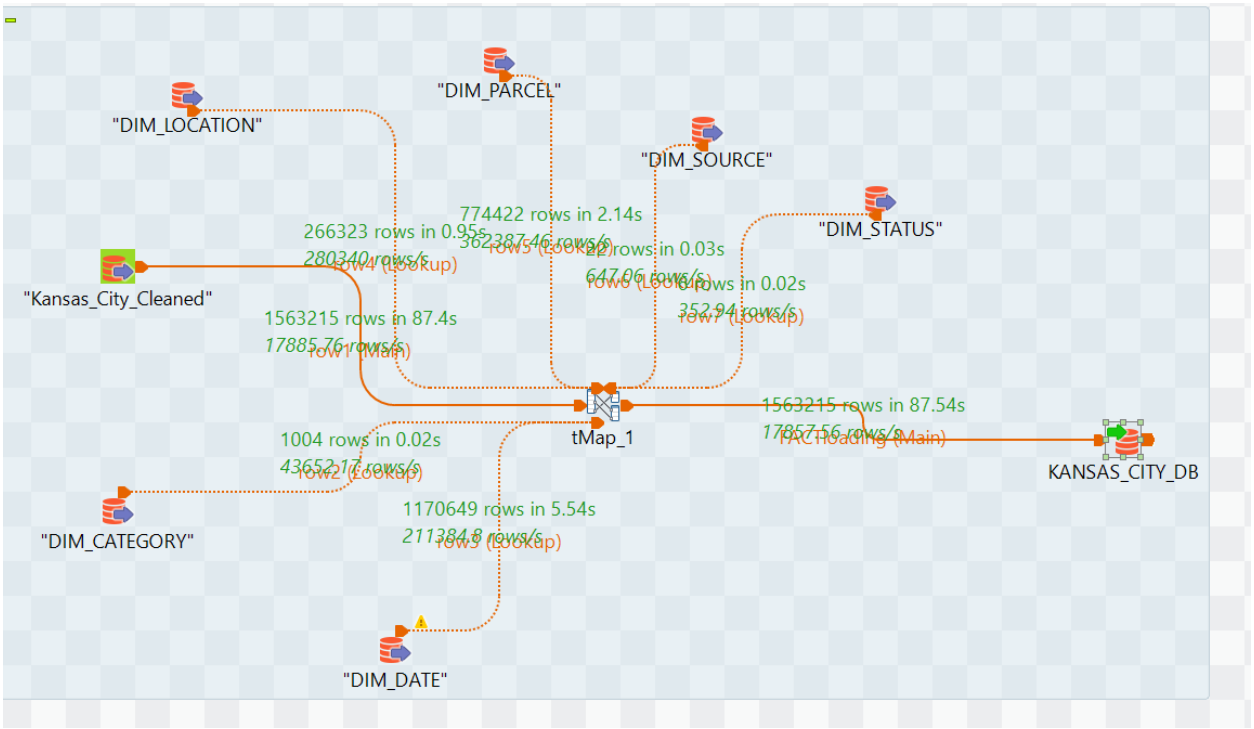
Dim date is derived by the same manner as well.

7.Dim_Status:



Dim status is derived by the same manner as well.

8.Fact_Case:



After creating 6 Dimension Table , we are creating Fact_Case table out of them.In the tmap , for the main pipeline, Kansas_city_cleaned table is inserted . From that table , all the other tables are mapped to it by the respective fields. Dim tables are connected to the tmap using lookup and each row is checked with the cleaned table. The row is pushed into the fact table if all the fields value in the dim and main table is equal.

Seasonal Trends:

A new calculated column named SEASON_NAME was added to the Dim Date table to categorize service requests by season based on the month of creation. This categorization helps in identifying seasonal patterns in service requests, allowing the city to anticipate and better manage fluctuations in demand.

Data Visualization and Sql Validations:

(1) Service Requests Over Time:

- What is the overall trend in Service Requests over the years 2018-2021?
- How have Service Requests changed on a monthly basis?

Enable Actual Plan | Parse | Enable SQLCMD | To Notebook

```sql
-- Service Requests change on a monthly basis
SELECT
    dd.[Creation_Calender_Year] AS Year,
    dd.[Creation_Month_Name] AS Month,
    COUNT(fc.[CASE ID]) AS Total_Service_Requests
FROM
    [Individual_Project_1].[dbo].[FACT_CASE] fc
JOIN
    [Individual_Project_1].[dbo].[DIM_DATE] dd
ON
    fc.[DateSK] = dd.[DateSK]
WHERE
    dd.[Creation_Calender_Year] BETWEEN 2018 AND 2021
GROUP BY
    dd.[Creation_Calender_Year],
    dd.[Creation_Month_Name],
    dd.[Creation_Month_Name]
ORDER BY
    dd.[Creation_Calender_Year],
    dd.[Creation_Month_Name];

```

Results | Messages

	Year	Month	Total_Service_Requests
1	2018	April	9657
2	2018	August	10960
3	2018	December	7959
4	2018	February	7728
5	2018	January	10870
6	2018	July	12329
7	2018	June	11644
8	2018	March	9625
9	2018	May	12295
10	2018	November	11419
11	2018	October	10095
12	2018	September	9699
13	2019	April	16108
14	2019	August	13207
15	2019	December	10122
16	2019	February	15596
17	2019	January	20641
18	2019	July	14194

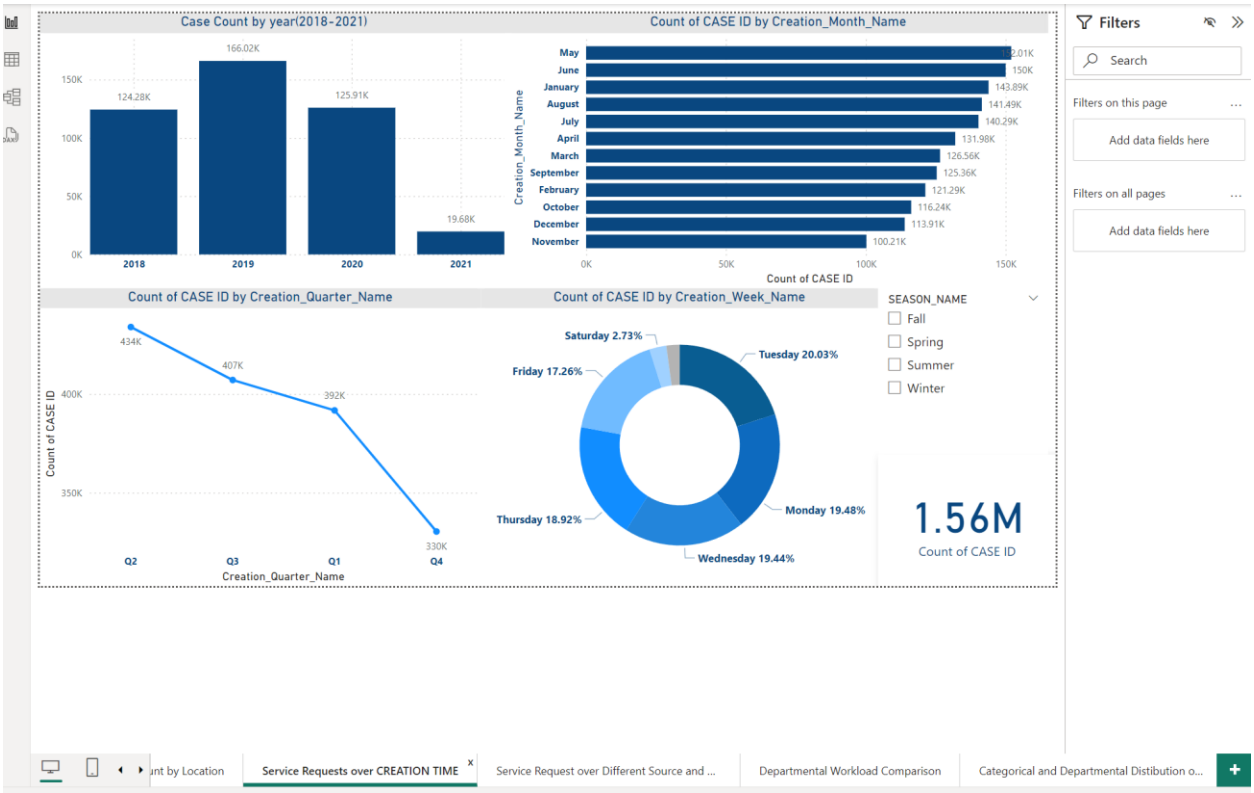
Enable Actual Plan | Parse | Enable SQLCMD | To Notebook

```sql
-- Overall trend in Service Requests over the years 2018-2021
SELECT
    dd.[Creation_Calender_Year] AS Year,
    COUNT(fc.[CASE ID]) AS Total_Service_Requests
FROM
    [Individual_Project_1].[dbo].[FACT_CASE] fc
JOIN
    [Individual_Project_1].[dbo].[DIM_DATE] dd
ON
    fc.[DateSK] = dd.[DateSK]
WHERE
    dd.[Creation_Calender_Year] BETWEEN 2018 AND 2021
GROUP BY
    dd.[Creation_Calender_Year]
ORDER BY
    dd.[Creation_Calender_Year];


```

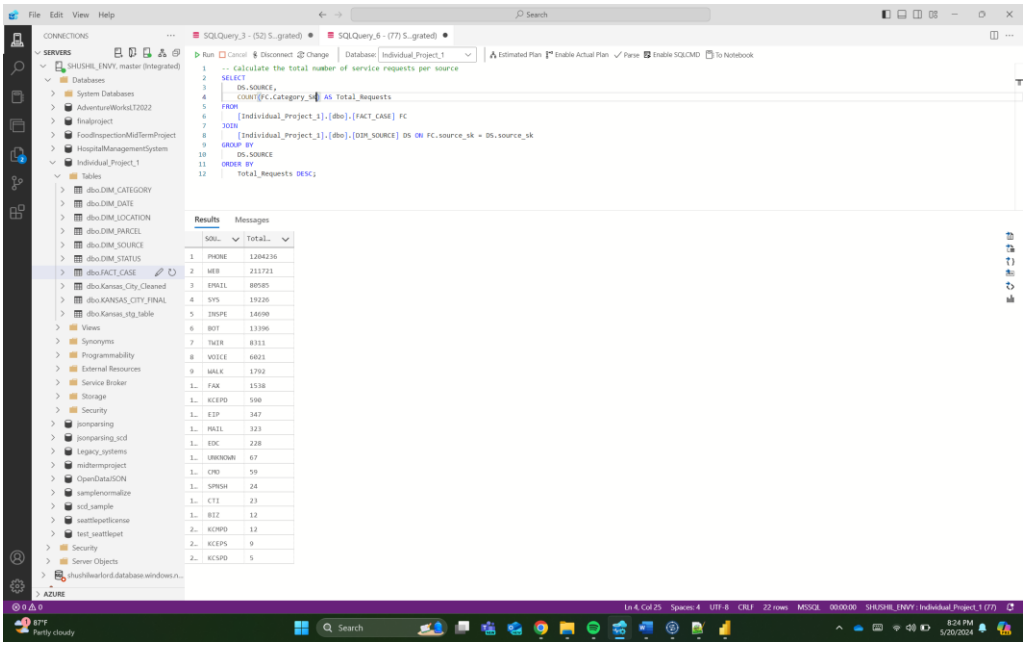
Results | Messages

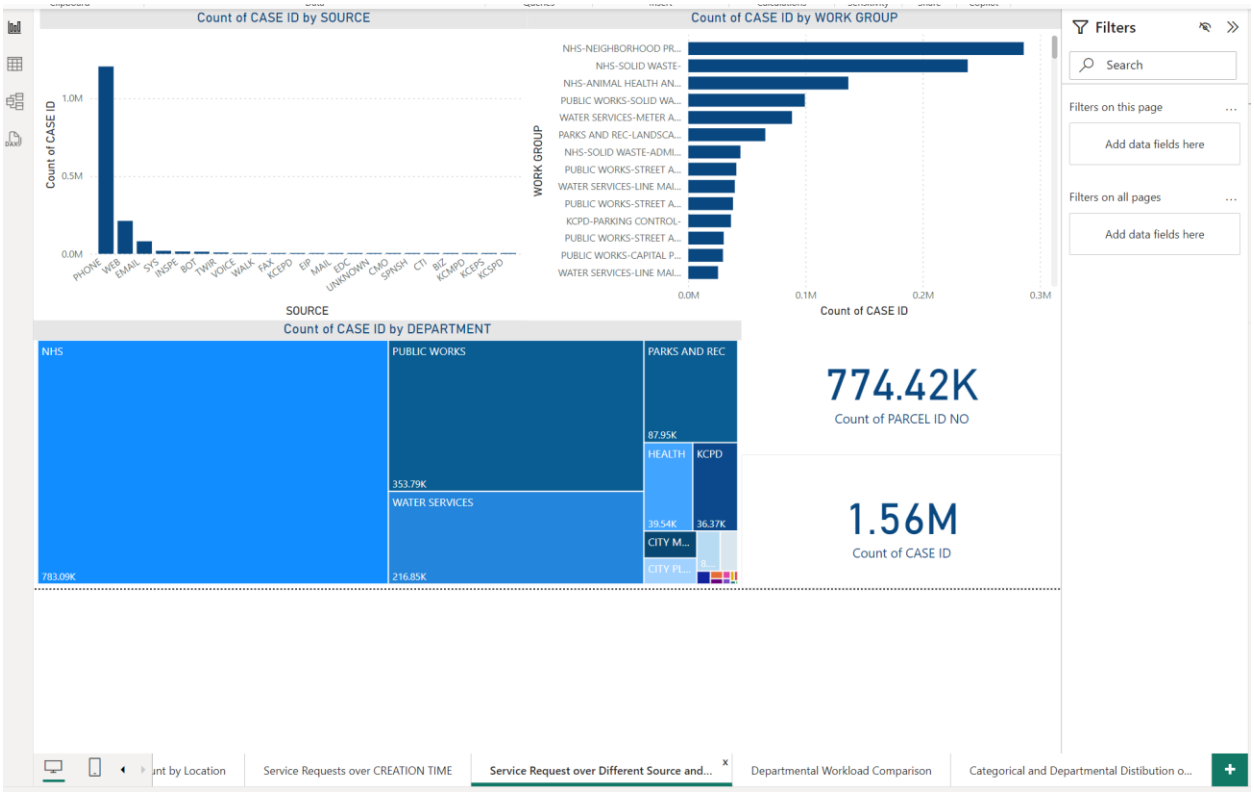
	Year	Total_Service_Requests
1	2018	124280
2	2019	166021
3	2020	125906
4	2021	19683



(2) Volume of service requests received from different sources:

• What is the overall trend in Service Requests over Sources?





(3) Volume of service requests received by Department:

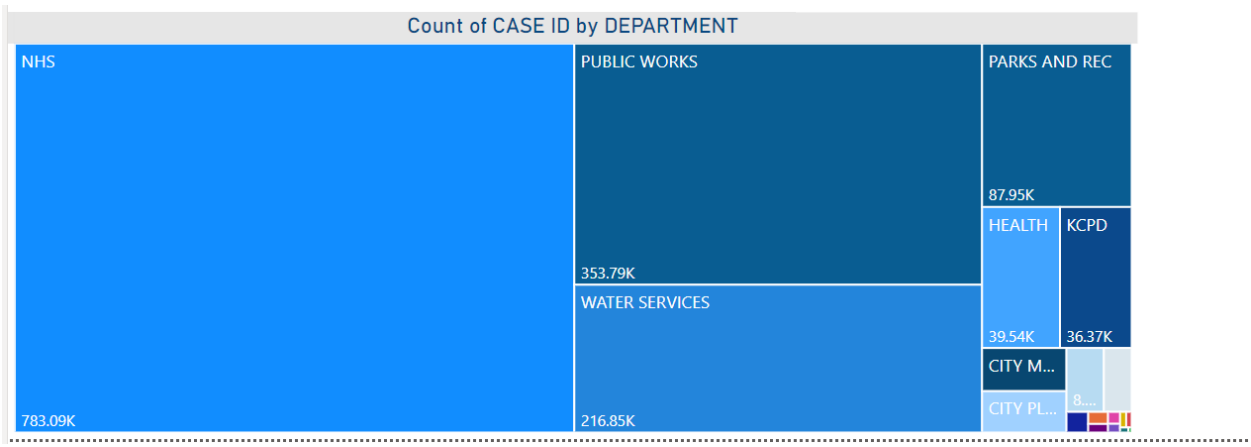
- What is the overall trend in Service Requests received by Departments?

```
    -- Calculate the total number of service requests per department
SELECT
    DP.DEPARTMENT,
    COUNT(FC.Category_SK) AS Total_Requests
FROM
    [Individual_Project_1].[dbo].[FACT_CASE] FC
JOIN
    [Individual_Project_1].[dbo].[DIM_PARCEL] DP ON FC.Parcel_SK = DP.Parcel_SK
GROUP BY
    DP.DEPARTMENT
ORDER BY
    Total_Requests DESC;
```

Results | Messages

	DEPARTMENT	Total_Requests
1	NHS	783094
2	PUBLIC WORKS	353787
3	WATER SERVICES	216852
4	PARKS AND REC	87954
5	HEALTH	39543
6	KCPD	36369
7	CITY MANAGERS OFFICE	13098
8	CITY PLANNING AND DEVELOPMENT	12575
9	NORTHLAND	8591
10	NCS	6391
11	FINANCE	1616
12	PARKS & REC	861
13	FIRE	612
14	GENERAL SERVICE	518
15	MUNICIPAL COURT	379
16	HOUSING COMMUNITY DEV	342
17	SOUTH	309
18	AVIATION	151
19	CONVENTION AND ENTERTAINMENT CENTER	59
20	MAYORS OFFICE	37
21	NORTHEAST	35
22	INFORMATION TECHNOLOGY	24
23	PARKS & RECREATION	11
24	CITY COUNCIL	3

Ln 25, Col 25 Spaces: 4 UTF-8 CRLF 27 rows MSSQL 00:00:00 SHUSHIL_ENVY : Individual_Project_1 (77)



(4) Top 10 Performance Metrics (Response Time) per CATEGORY and Type of Request:

• What are the top 10 cases whose response time was fastest? Categorize it with Category1 and Type of Request.

```sql
-- Calculate the response time for each case and categorize by Category1
SELECT TOP 10
    cat.CATEGORY1,
    CAST(AVG(CAST(fc.DAYS_TO_CLOSE AS DECIMAL(10, 2))) AS DECIMAL(10, 2)) AS average_response_time
FROM
    FACT_CASE fc
JOIN
    DIM_CATEGORY cat ON fc.Category_SK = cat.Category_SK
GROUP BY
    cat.CATEGORY1
ORDER BY
    average_response_time ASC;
```

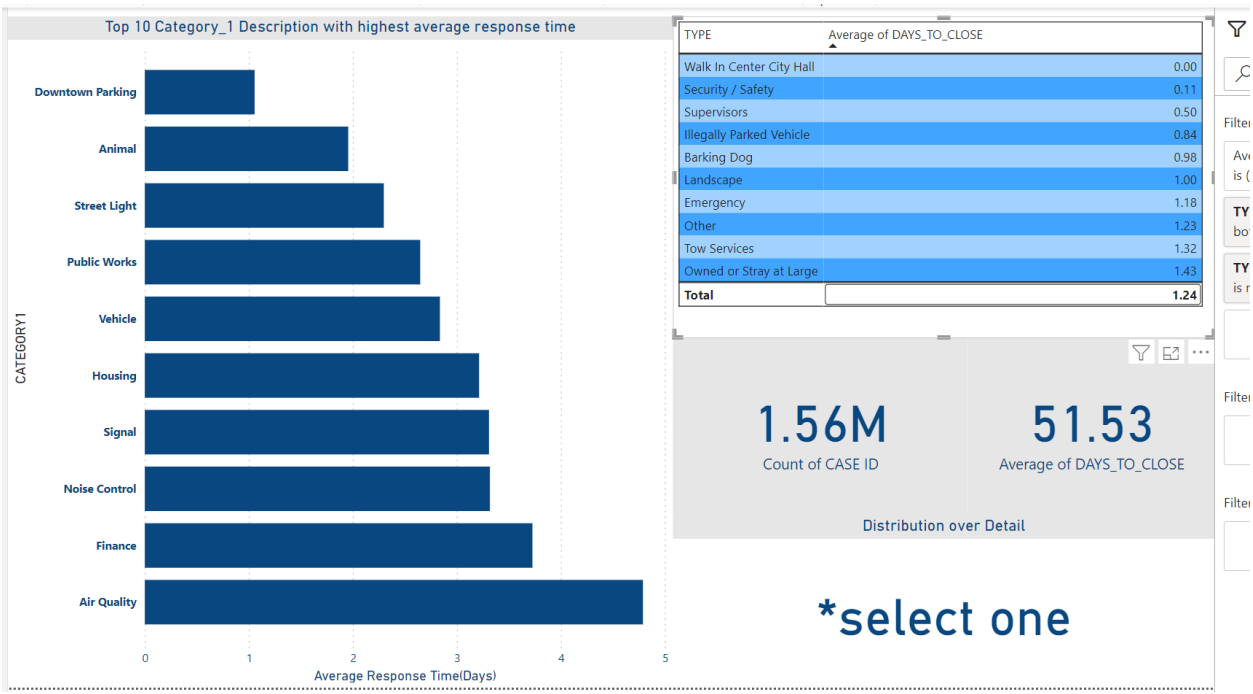
Results | Messages

CATEGORY1	average_response_time
Downtown Parking	1.06
Animal	1.96
Street Light	2.30
Public Works	2.65
Vehicle	2.84
Housing	3.21
Signal	3.31
Noise Control	3.32
Finance	3.73
Air Quality	4.79

```sql
-- Calculate the response time for each case by type of requests
SELECT TOP 10
    cat.TYPE,
    CAST(AVG(CAST(fc.DAYS_TO_CLOSE AS DECIMAL(10, 2))) AS DECIMAL(10, 2)) AS average_response_time
FROM
    FACT_CASE fc
JOIN
    DIM_CATEGORY cat ON fc.Category_SK = cat.Category_SK
GROUP BY
    cat.TYPE
ORDER BY
    average_response_time ASC;
```

Results | Messages

	TYPE	average_response_time
1	Walk In Center City Hall	0.00
2	Security / Safety	0.11
3	Supervisors	0.50
4	Illegally Parked Vehicle	0.84
5	Barking Dog	0.98
6	Landscape	1.00
7	Emergency	1.18
8	Other	1.23
9	Tow Services	1.32
10	Owned or Stray at Large	1.43



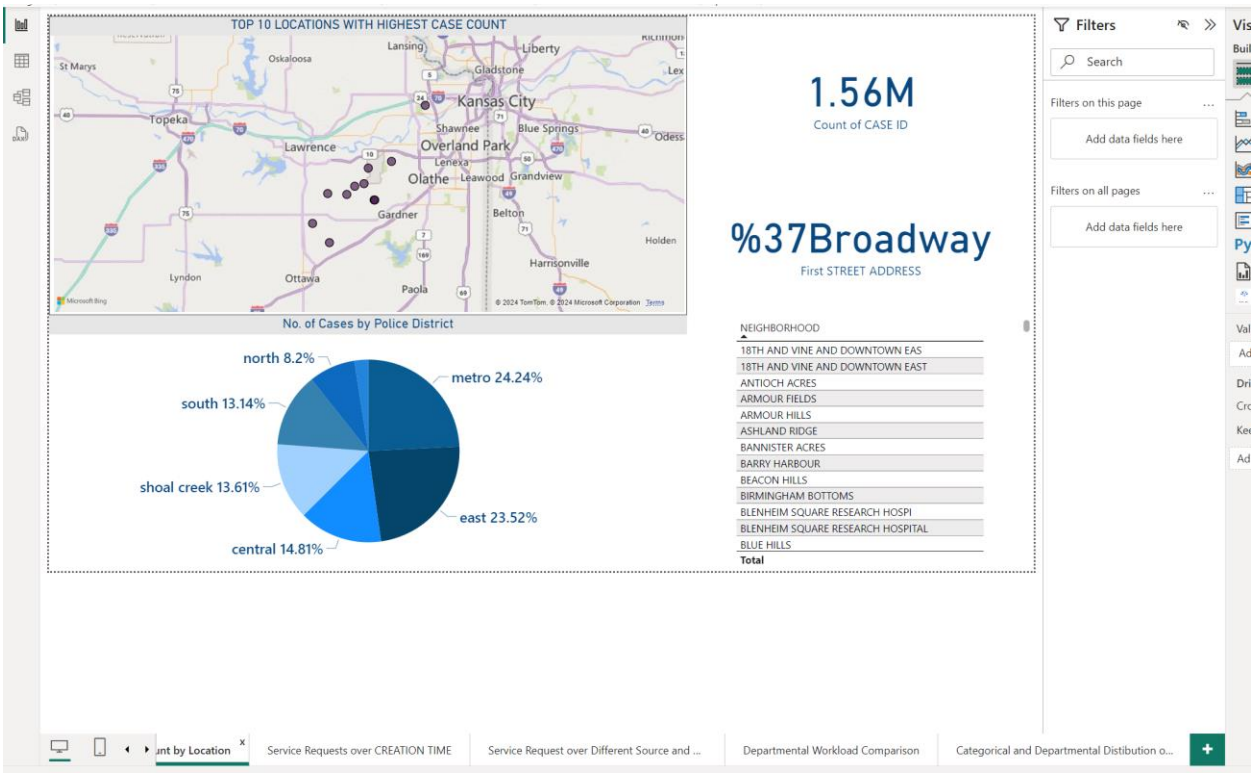
(5) Geographical Visualization:

- What are the Top 10 areas where most number of request were raised?

```
67
68  select top 10 l.[ZIP CODE],count(fc.Category_SK) as Casecount
69  from FACT_CASE fc join DIM_LOCATION l on fc.Location_SK=l.Location_SK
70   GROUP by l.[ZIP CODE]
71   order by Casecount desc
72
73
74
```

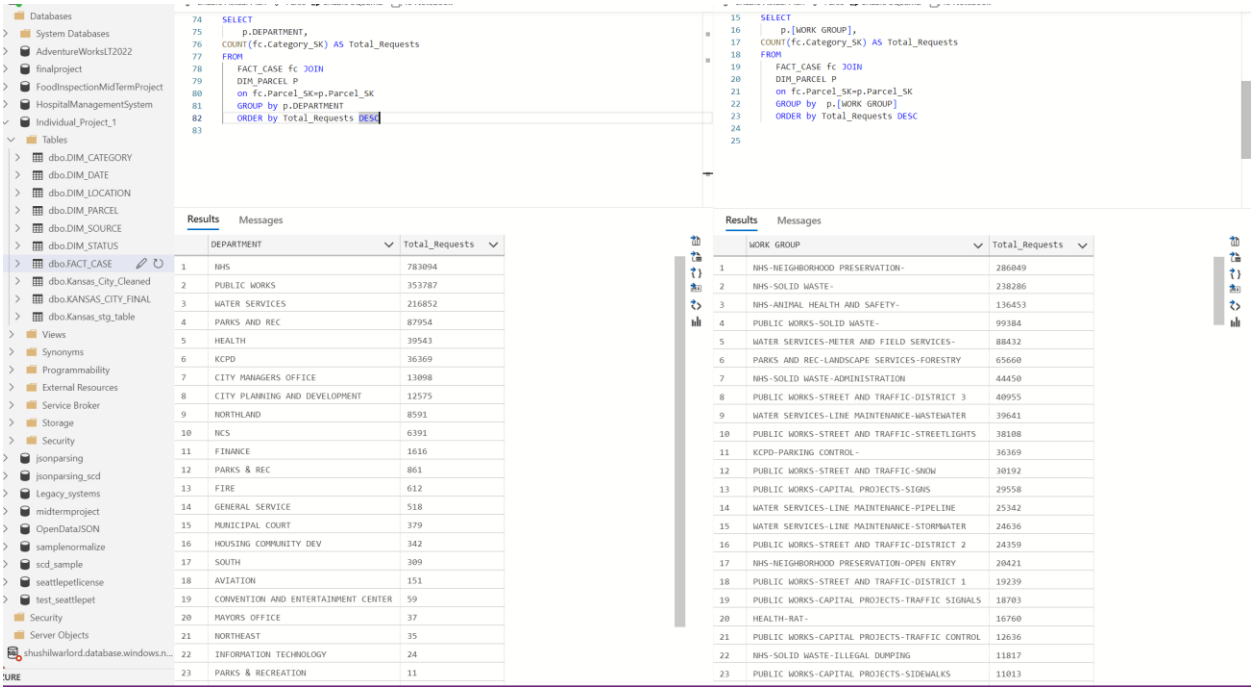
Results Messages

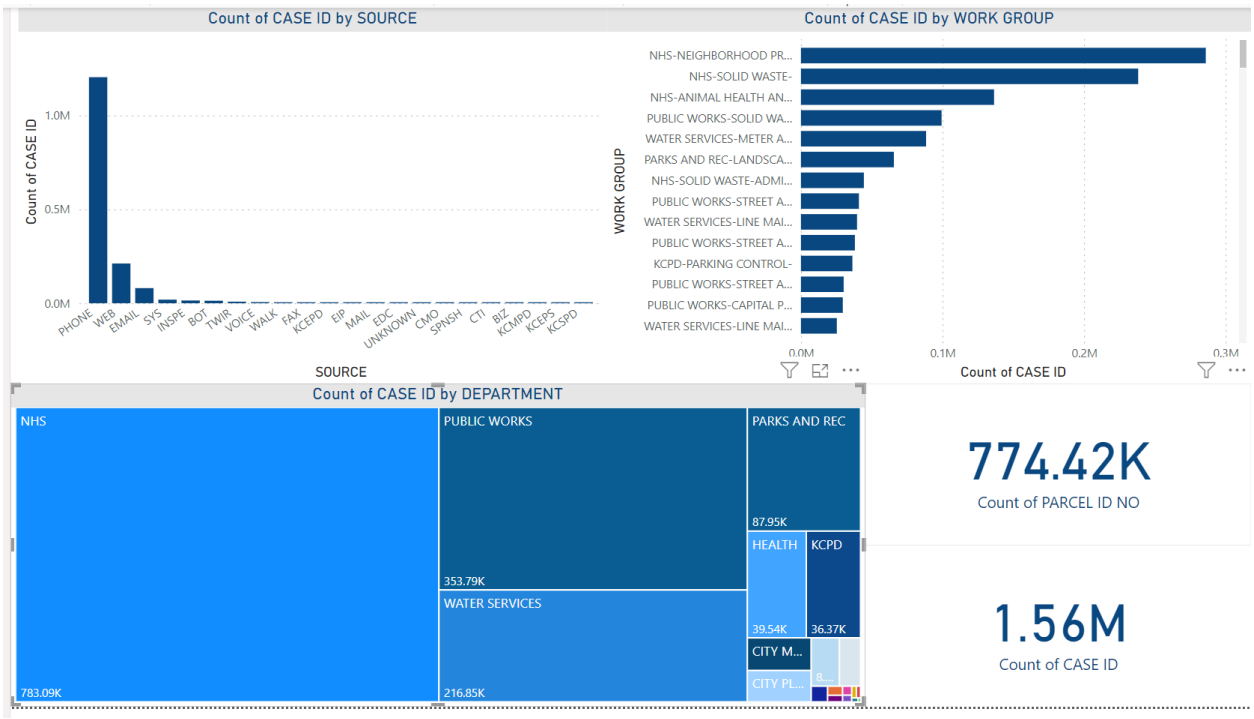
	ZIP CODE	Casecount
1	64130	118857
2	64127	80877
3	64114	74918
4	64134	71696
5	64132	69553
6	64131	69322
7	64128	65321
8	64110	57884
9	64119	51785
10	64111	45657



(6) Departmental Workload Comparison:

• How does the workload vary among different departments and work groups? Create a visual representation to highlight the distribution.





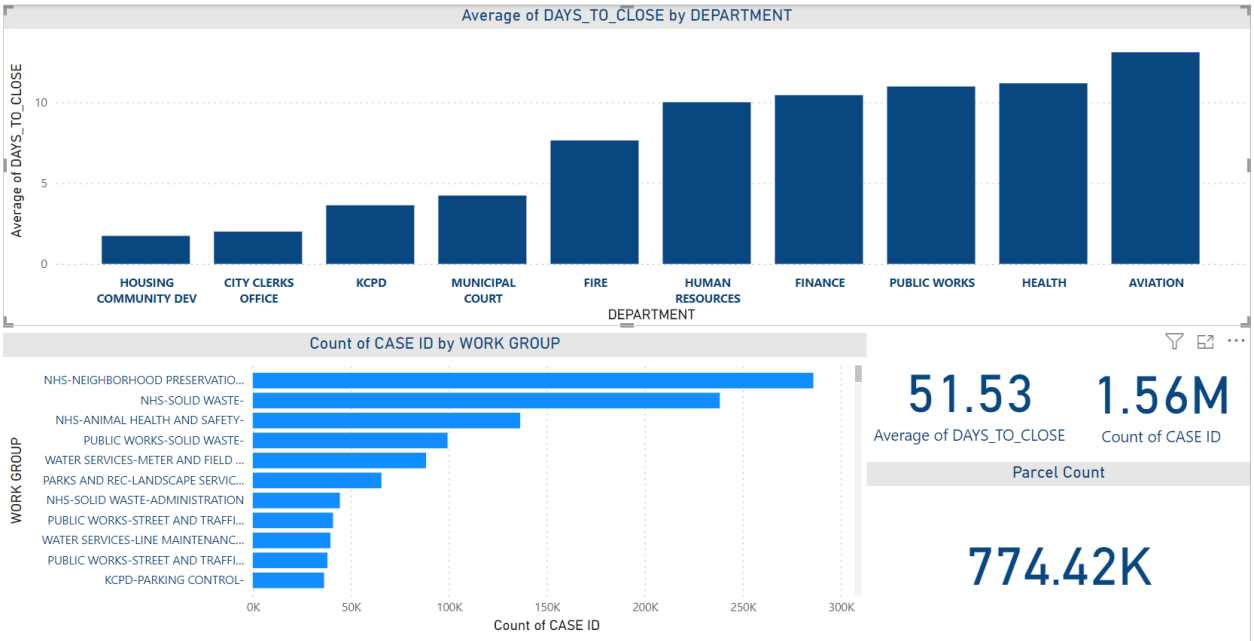
(7) Response Time Analysis:

• Visualize the distribution of response times for each department. Are there any outliers or patterns in response times?

```
40  -- Calculate the response time for each case and categorize by Deparment
41  SELECT TOP 10
42      p.DEPARTMENT,
43      CAST(AVG(CAST(fc.DAYS_TO_CLOSE AS DECIMAL(10, 2))) AS DECIMAL(10, 2)) AS average_response_time
44  FROM
45      FACT_CASE fc
46  JOIN
47      DIM_PARCEL p ON fc.Parcel_SK = p.Parcel_SK
48  GROUP BY
49      p.DEPARTMENT
50  ORDER BY
51      average_response_time ASC;
52
```

Results Messages

	DEPARTMENT	average_response_time
1	HOUSING COMMUNITY DEV	1.73
2	CITY CLERKS OFFICE	2.00
3	KCPD	3.63
4	MUNICIPAL COURT	4.23
5	FIRE	7.64
6	HUMAN RESOURCES	10.00
7	FINANCE	10.44
8	PUBLIC WORKS	10.98
9	HEALTH	11.17
10	AVIATION	13.09



Days_to_Close is inversely proportional to response time .So smaller the average of it , faster the response time.

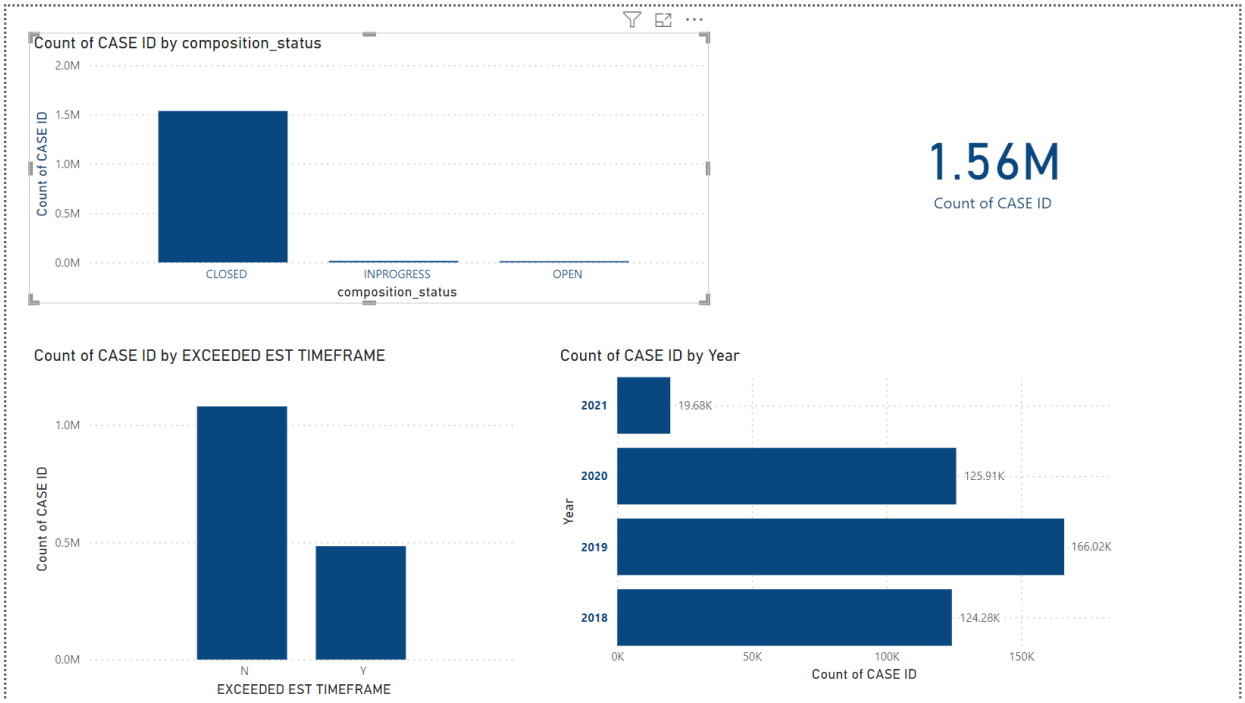
(8) Service Request Status Composition:

• Create a visualization to show the composition of service request statuses (open, closed, in progress). How has this composition changed over the years 2018-2021?

```
24  --Composition of Service requests over status_names aggreated values into open,close and resolved
25  WITH DerivedStatus AS (
26      SELECT s.[status_SK],
27          CASE
28              WHEN [Status] IN ('resolved', 'closed') THEN 'closed'
29              WHEN [Status] = 'open' THEN 'open'
30              ELSE 'inprogress'
31          END AS StatusDerived
32      FROM
33          DIM_STATUS s join FACT_CASE c on s.status_SK=c.status_SK
34  )
35
36  SELECT
37      StatusDerived,
38      COUNT(s1.status_SK) AS CaseCount
39  FROM
40      DerivedStatus s1
41  GROUP BY
42      StatusDerived;
43
44
```

Results Messages

	StatusDerived	CaseCount
1	closed	1535389
2	inprogress	15169
3	open	12657



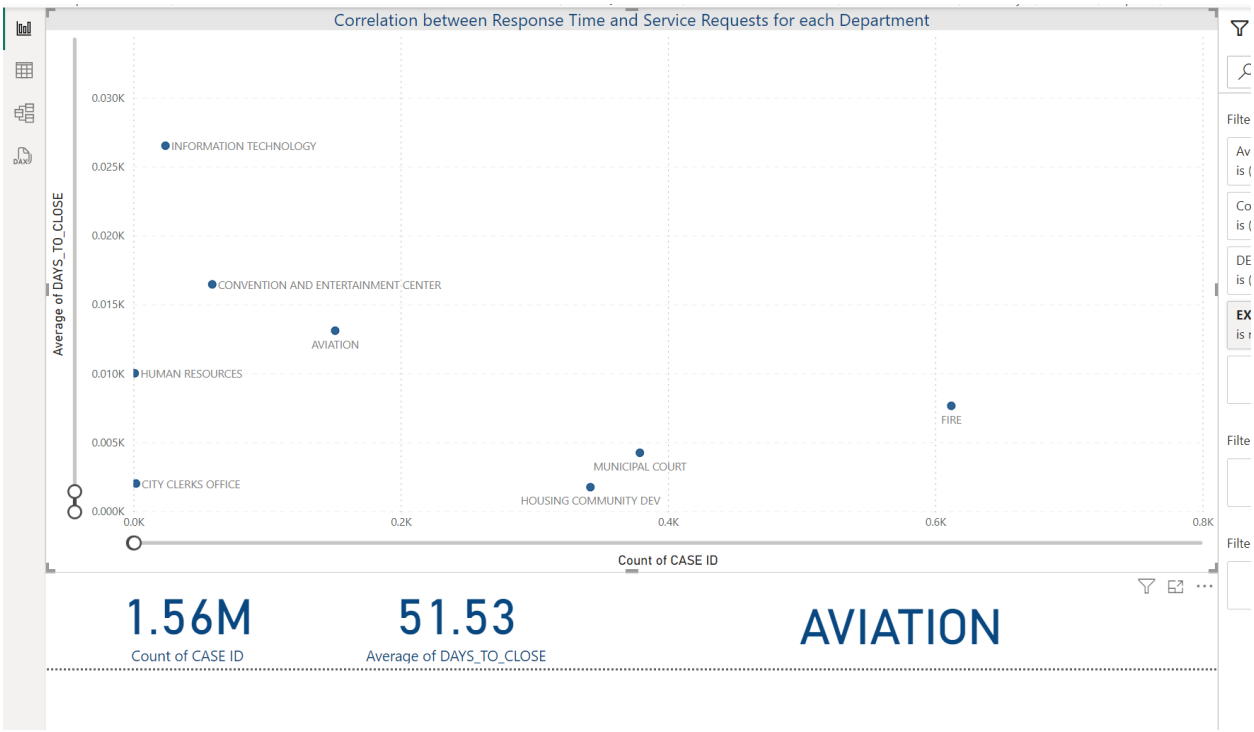
(9) Time to Closure Analysis:

• Visualize the average days to close service requests for each category1. Are there categories with consistently longer closure times?

• Show top 10 (If you need help on how to restrict top 10 contact us and we can guide / help you) Same as **4)**

(10) Workload Efficiency:

• Create a visualization to show the relationship between workload (number of service requests) and efficiency (days to close) for each department?



Source Optimization:

Analyzing the distribution of service request sources reveals that phone calls are the predominant mode of communication, followed by web and email submissions. This insight can guide resource allocation to ensure that the most commonly used channels are well-supported and optimized for efficiency.

Geographic Analysis:

The geographic data, including street addresses, ZIP codes, neighborhoods, and police districts, allows for a granular analysis of service request hotspots. Identifying these areas enables targeted interventions and resource deployment, ensuring that high-demand areas receive appropriate attention.

Insights and Recommendations

Seasonal Trends: The analysis of seasonal trends can inform resource planning and allocation. For example, if a higher volume of requests is observed during certain seasons, the city can proactively allocate additional resources during these periods to maintain service levels.

Source Optimization: Understanding the predominant sources of service requests can help the city enhance its service delivery channels. For instance, improving the efficiency of phone and web-based request handling could significantly reduce response times and improve citizen satisfaction.

Geographic Targeting: By identifying geographic hotspots for service requests, the city can prioritize interventions and allocate resources more effectively. This targeted approach ensures that areas with higher service demand are adequately supported, improving overall service quality.

Conclusion

This analysis of the 311 service request data provides a detailed understanding of the dataset's structure, quality, and key insights. By leveraging data profiling and dimensional modeling, Kansas City can enhance its operational efficiency, improve resource allocation, and ultimately provide better services to its residents. As the city transitions to a new record management system, continued monitoring and integration of the new data will ensure seamless service delivery and ongoing improvements in citizen satisfaction.