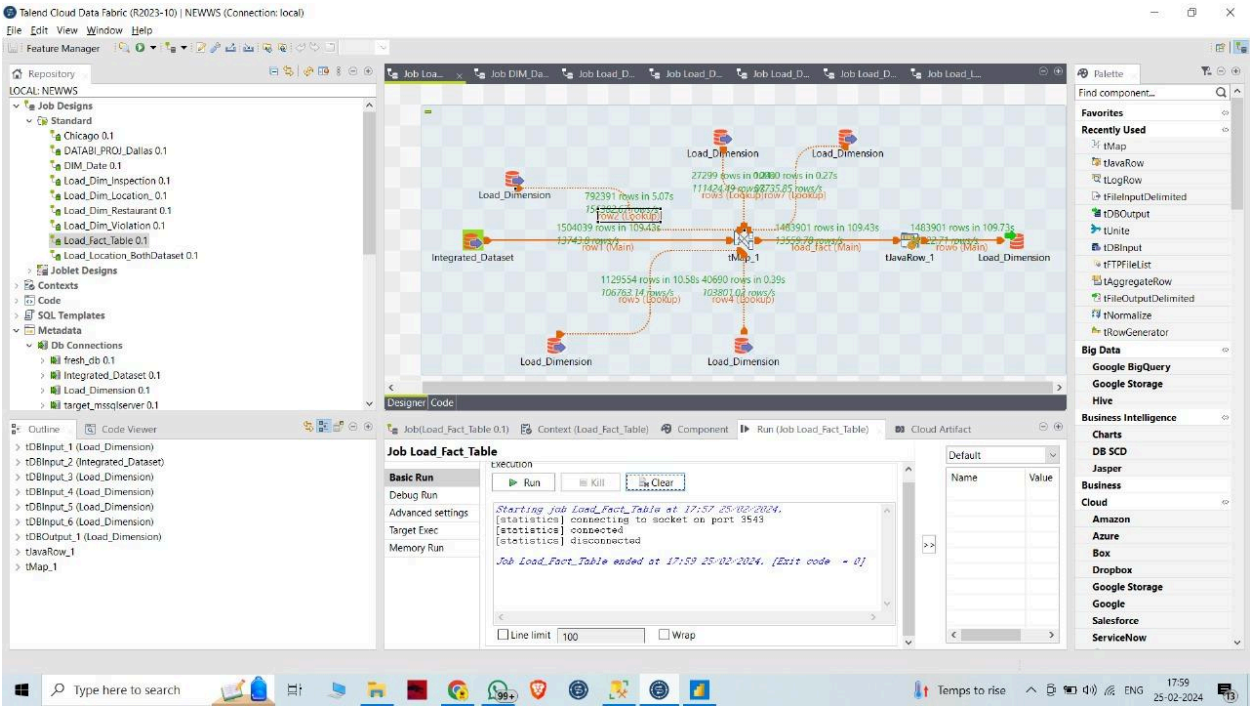
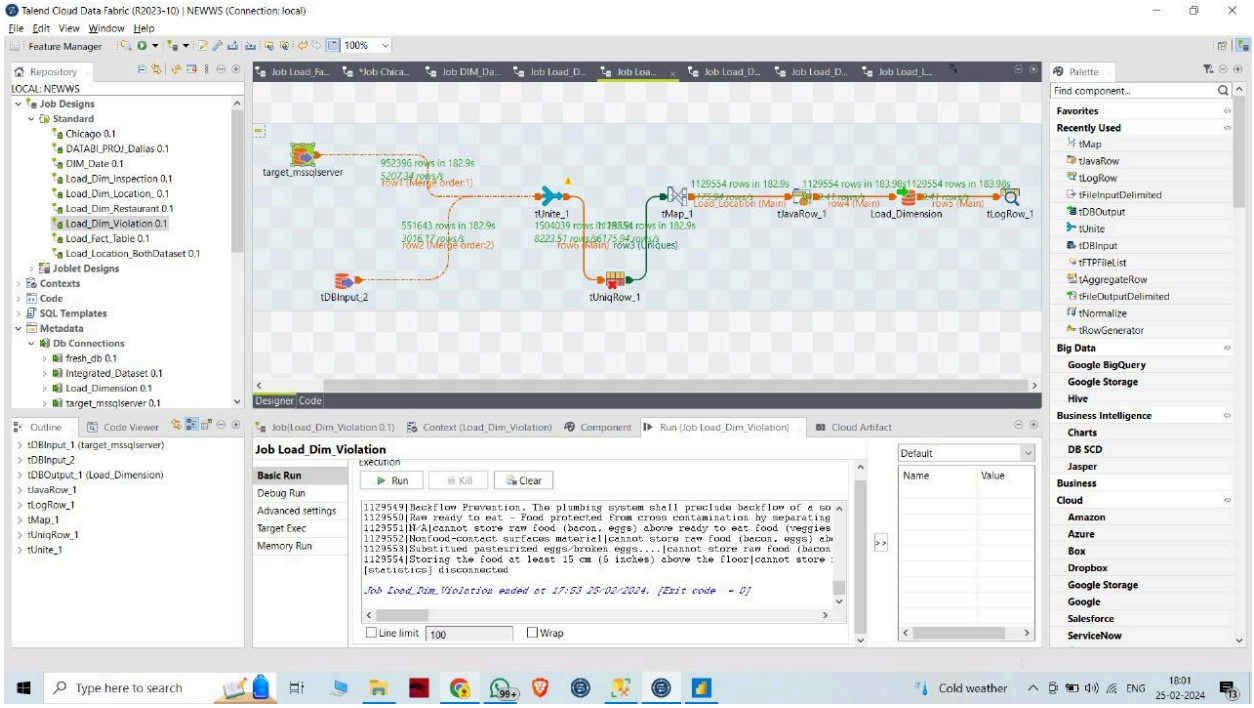
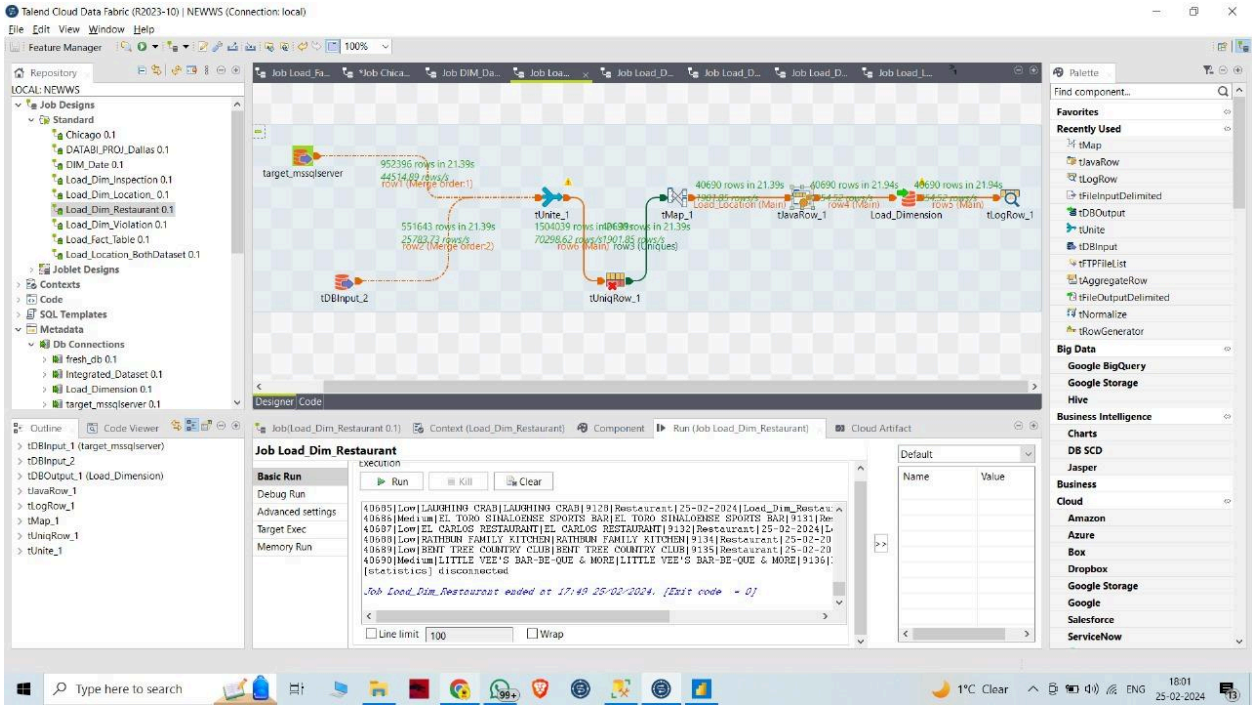
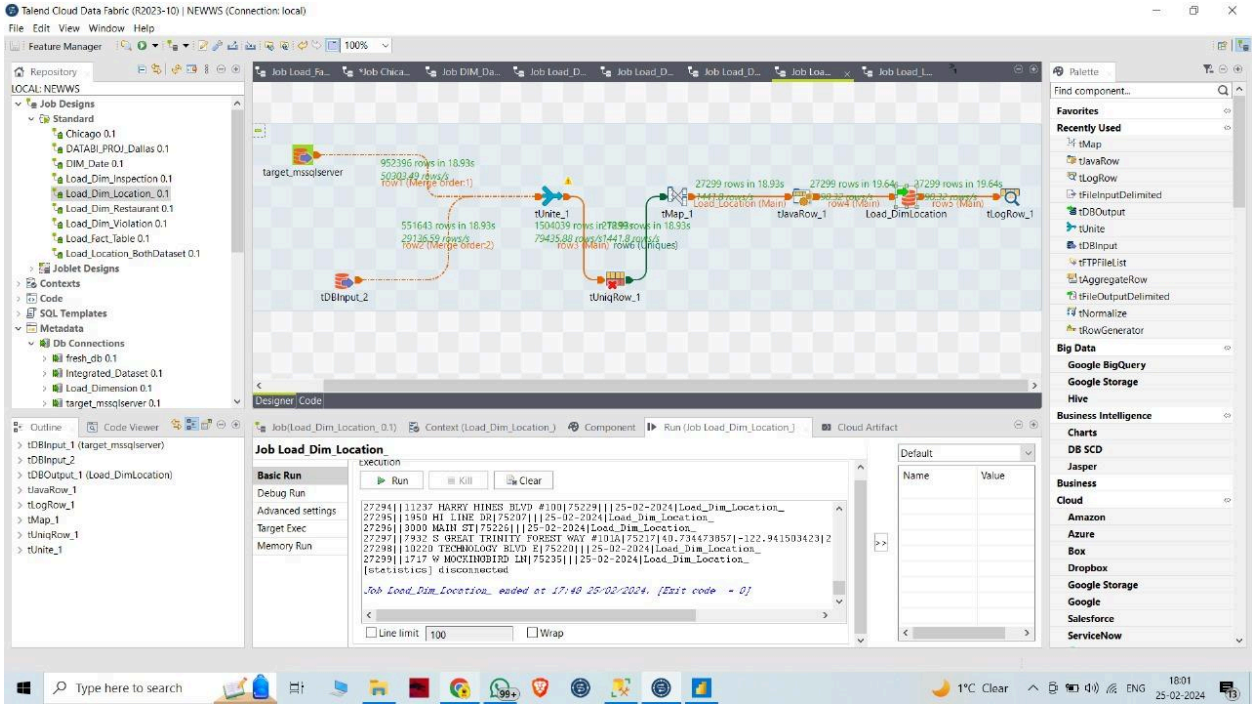
PART 1

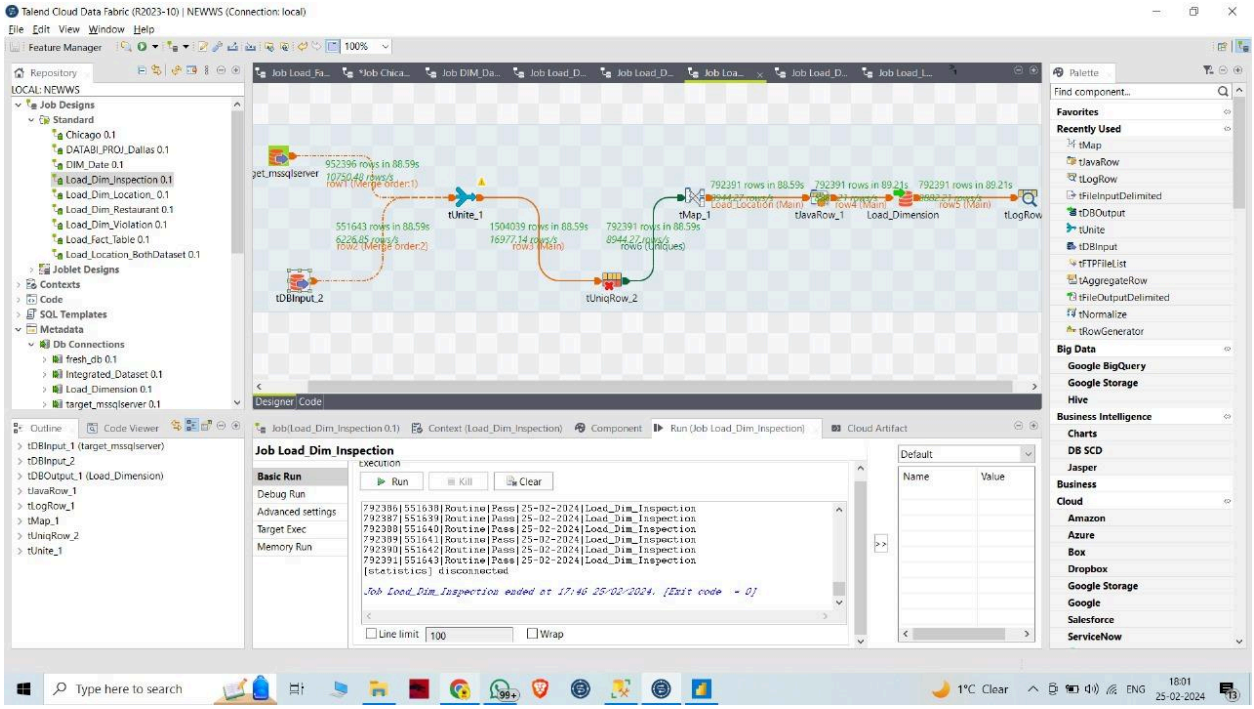
Screenshot of workflows completed

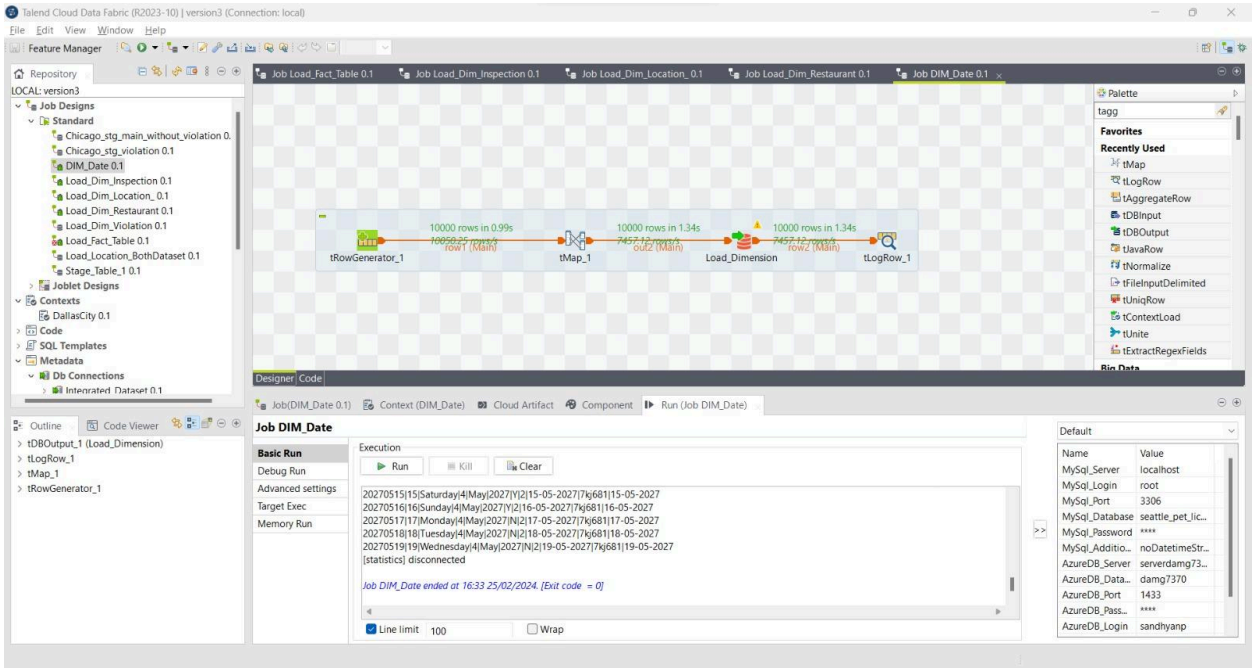



Talend Cloud Data Fabric (R2023-10) | NEWWS (Connection: local)
File Edit View Window Help
Job Load_Dim_Violation
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Run
Clear
Job Load_Dim_Violation ended at 17:53 25/02/2024. [Exit code = 0]
Type here to search
Cold weather
ENG
18:01
25-02-2024


Talend Cloud Data Fabric (R2023-10) | NEWWS (Connection: local)
File Edit View Window Help
Job Load_Dim_Restaurant
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Run
Clear
Job Load_Dim_Restaurant ended at 17:49 25/02/2024. [Exit code = 0]
Type here to search
1°C Clear
ENG
18:01
25-02-2024


Talend Cloud Data Fabric (R2023-10) | NEWWS (Connection: local)
File Edit View Window Help
Job Load_Dim_Location_
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Run
Kill
Clear
27294||11237 HARRY HINES BLVD #100|75229|||25-02-2024|Load_Dim_Location_
27295||1953 HI LINE DR|75207|||25-02-2024|Load_Dim_Location_
27296||3000 MAIN ST|75226|||25-02-2024|Load_Dim_Location_
27297||7932 S GREAT TRINITY FOREST WAY #101A|75217|40.734473857|-122.941503423|2
27298||10220 TECHNOLOGY BLVD E|75220|||25-02-2024|Load_Dim_Location_
27299||1717 W MOCKINGBIRD LN|75235|||25-02-2024|Load_Dim_Location_
[statistics] disconnected
Job Load_Dim_Location_ ended at 17:48 25-02-2024. [Exit code = 0]
Line limit 100
Wrap
Default
Name
Value
1°C Clear
ENG
18:01
25-02-2024


Talend Cloud Data Fabric (R2023-10) | NEWWS (Connection: local)
File Edit View Window Help
Job Load_Dim_Inspection
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Run
Kill
Clear
792386|551638|Routine|Pass|25-02-2024|Load_Dim_Inspection
792387|551639|Routine|Pass|25-02-2024|Load_Dim_Inspection
792388|551640|Routine|Pass|25-02-2024|Load_Dim_Inspection
792389|551641|Routine|Pass|25-02-2024|Load_Dim_Inspection
792390|551642|Routine|Pass|25-02-2024|Load_Dim_Inspection
792391|551643|Routine|Pass|25-02-2024|Load_Dim_Inspection
[statistics] disconnected
Job Load_Dim_Inspection ended at 17:46 25-02-2024. [Exit code = 0]
Line limit 100
Wrap
Default
Name
Value
1°C Clear
ENG
18:01
25-02-2024



DDL SCRIPT FOR STAGE TABLE:

```
USE [chicago]
GO

/** Object: Table [dbo].[stg_table_Chicago_main_without_violation] Script Date: 2/25/2024 4:42:22 PM **/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[stg_table_Chicago_main_without_violation](
	[Inspection_ID] [int] NULL,
```

```
	[DBA_Name] [varchar](100) NULL,
	[AKA_Name] [varchar](100) NULL,
	[Licence] [int] NULL,
	[Facility_Type] [varchar](100) NULL,
	[Address] [varchar](100) NULL,
	[City] [varchar](100) NULL,
	[Zip_Code] [int] NULL,
	[Inspection_Date] [datetime] NULL,
	[Inspection_Type] [varchar](1000) NULL,
	[Latitude] [varchar](100) NULL,
	[Longitude] [varchar](100) NULL,
	[Risk] [varchar](100) NULL,
	[DI_CreateDate] [datetime] NULL,
	[DI_WorkFlowFileName] [varchar](100) NULL
) ON [PRIMARY]
GO
USE [chicago]
GO

/** Object:  Table [dbo].[stg_table_Chicago_violation]    Script Date: 2/25/2024
4:42:43 PM **/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[stg_table_Chicago_violation](
	[Violation_Description] [varchar](4574) NULL,
	[Violation_Comments] [varchar](max) NULL,
	[Violation_Code] [varchar](100) NULL,
	[Inspection_ID] [int] NULL,
	[Inspection_Type] [varchar](100) NULL,
	[Risk] [varchar](15) NULL,
	[Results] [varchar](100) NULL,
	[DI_CreateDate] [datetime] NULL,
	[DI_WorkFlowFileName] [varchar](100) NULL,
```

```
	[TotalViolations] [int] NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
 USE [poc]
GO

/** Object:  Table [dbo].[stg_Dallas_Main]    Script Date: 2/25/2024 4:43:01 PM **/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[stg_Dallas_Main](
	[Inspection_ID] [int] NULL,
	[DBA_Name] [varchar](500) NULL,
	[AKA_Name] [varchar](500) NULL,
	[Licence] [int] NULL,
	[Facility_Type] [varchar](60) NULL,
	[Address] [varchar](100) NULL,
	[City] [varchar](10) NULL,
	[Zip_Code] [int] NULL,
	[Inspection_Date] [datetime] NULL,
	[Inspection_Type] [varchar](60) NULL,
	[Latitude] [varchar](60) NULL,
	[Longitude] [varchar](52) NULL,
	[Risk] [varchar](100) NULL,
	[DI_CreateDate] [datetime] NULL,
	[DI_WorkFlowFileName] [varchar](60) NULL
) ON [PRIMARY]
GO
 USE [poc]
GO

/** Object:  Table [dbo].[stg_Dallas_Violation]    Script Date: 2/25/2024 4:43:18 PM **/
SET ANSI_NULLS ON
```

```
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[stg_Dallas_Violation](
	[Violation_Description] [varchar](max) NULL,
	[Violation_Comments] [varchar](max) NULL,
	[Violation_Points] [int] NULL,
	[Violation_Code] [varchar](10) NULL,
	[Inspection_ID] [int] NULL,
	[Inspection_Type] [varchar](1000) NULL,
	[Risk] [varchar](50) NULL,
	[DI_CreatedDate] [datetime] NULL,
	[DI_WorkFlowFileName] [varchar](60) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
 USE [DDLScript]
GO

/** Object:  Table [dbo].[stg_Integrated_Main_Dataset]    Script Date: 2/25/2024
4:43:50 PM **/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[stg_Integrated_Main_Dataset](
	[DBA_Name] [varchar](1000) NULL,
	[AKA_Name] [varchar](100) NULL,
	[Facility_Type] [varchar](60) NULL,
	[Address] [varchar](1000) NULL,
	[Zipcode] [int] NULL,
	[Inspection_Date] [datetime] NULL,
	[Inspection_Type] [varchar](50) NULL,
	[Latitude] [varchar](100) NULL,
```

```
    [Longitude] [varchar](100) NULL,
    [Inspection_ID] [int] NULL,
    [Licence] [int] NULL,
    [Risk] [varchar](100) NULL,
    [DI_CreateDate] [datetime] NULL,
    [DI_WorkFlowFileName] [varchar](1000) NULL
) ON [PRIMARY]
GO
```

Analysis report on Data profiling:

CHICAGO:

Overview and Dataset Statistics:

Variables: The dataset contains 17 variables across different data types, including numeric, text, categorical, and DateTime.

Observations: There are 267,603 observations in the dataset.

Missing Cells: Out of the total cells in the dataset, 84,092 are missing, accounting for 1.8% of all cells.

Duplicate Rows: There are no duplicate rows, ensuring data uniqueness.

Memory Usage: The dataset occupies 34.7 MiB in memory, with an average record size of 136.0 bytes.

Variable Types:

Numeric: 5 variables

Text: 8 variables

Categorical: 3 variables

DateTime: 1 variable

Key Variables Analysis:

Inspection ID: Unique across all observations (267,603 distinct values), indicating it serves well as a primary key.

DBA Name and AKA Name: Represent the official and alternative names of establishments with some missing values in "AKA Name".

License : Shows some missing values and zeros, indicating unlicensed or possibly closed establishments.

Facility Type: Contains 5,114 missing values, highlighting a need for better classification or data entry processes.

Risk: Categorized into three levels with minimal missing values, providing insights into inspection priorities.

Violations: A significant portion of the dataset (27.4% missing) lacks violation details, suggesting either compliance or missing documentation.

Missing Values :

Missing Values: The missing data across various variables like "AKA Name", "License #", "Facility Type", and especially "Violations" pose challenges for analysis, requiring imputation or exclusion strategies depending on the analysis goals.

DALLAS:

General Overview:

Variables: There are 114 variables, covering a wide range of information from basic establishment details to extensive violation descriptions.
Observations: The dataset comprises 78,400 records.
Missing Data: A significant portion of the dataset, 72.2% or 6,454,357 cells, are missing, indicating substantial gaps in the recorded information.
Duplicates: There are 42 duplicate rows, constituting a minor 0.1% of the dataset.
Memory Usage: The dataset occupies 68.2 MiB in memory, with an average record size of 912.0 bytes.
Key Variable Insights:
Restaurant Name: Mostly unique with 9,136 distinct names, indicating a wide variety of establishments inspected.
Inspection Type: Shows an imbalance with three distinct categories, dominated by routine inspections.
Inspection Score: Varies widely from -26 to 100, with an average score of 90.87, suggesting generally high compliance levels.
Street Direction and Street Unit: These fields have significant missing data, 67% and 64.4% respectively, which could impact location-specific analyses.
Violation Descriptions and Points: Detailed across multiple columns, indicating a structured approach to documenting specific issues found during inspections. However, there's a notable percentage of missing data across these variables, particularly as the violation number increases, indicating not all establishments have violations or not all information is captured.

Data Quality Concerns:

High Percentage of Missing Data: The extensive missing information, especially in the violation-related variables, poses challenges for comprehensive analysis and may require imputation or exclusion strategies.
Duplicate Rows: The presence of duplicates, though a small fraction, necessitates cleaning to ensure accuracy in analysis and reporting.

The analysis of the Chicago dataset reveals several observations and potential issues with the data:
1. Missing Values :

There are missing values in several columns which could impact data analysis or model training processes. Specifically:

- AKA Name: 2,471 missing values.
- Licence : 18 missing values.
- Facility Type: 5,119 missing values.
- Risk: 82 missing values.
- City: 161 missing values.
- State: 59 missing values.
- Zip: 49 missing values.
- Inspection Type: 1 missing value.
- Violations: 73,407 missing values.
- Latitude, Longitude, and Location: 925 missing values.

Handling Missing Values :

● Imputation : For columns like AKA Name, Facility Type, and Risk, consider using the most frequent value or a placeholder value (e.g., "Unknown") to fill missing entries, especially if these fields are categorical and used in analyses.
● Deletion : If the missing values constitute a small fraction of the dataset and are missing completely at random, consider removing these rows. However, this approach might not be suitable for Violations due to the large number of missing values.

2. <u>Data Types and Conversions</u> :

- The Licence # and Zip columns are represented as floating-point numbers, which may not be appropriate since these are categorical identifiers and should be treated as Strings.
- Inspection Date is in object (String) format. For any time series analysis or operations that require date manipulation, this column should be converted to Datetime format. Inspection_ID is converted to integer.

Handling Data Types and Conversion :
● Conversion to Appropriate Types : Convert Licence # and Zip to strings using .astype(str) to prevent any mathematical operations on these identifiers. The Inspection Date should be converted to a datetime format using date datatype for easier manipulation and analysis over time. Inspection_ID is converted to integer

3. <u>Duplicates and Uniqueness</u> :

- While a detailed check for duplicate rows was not explicitly performed here, the uniqueness of certain identifiers like Inspection ID suggests that each row represents a unique inspection event.

- The Violations column has a high number of unique values (193,077), but given the nature of this data (text descriptions of violations), this is expected. Still, it indicates that there might be a lot of text-based data that needs to be carefully processed or standardised for analysis.
-A redundant column exists, such as "DBA Name" and "AKA Name," containing restaurant names that are identical in both columns.

Handling Duplicates and Uniqueness :

● Identifying and Removing Duplicates : Use DataFrame.duplicated(to find and DataFrame.drop_duplicates() to remove duplicate rows, if any. The transformation was carried out within Tmap

4. <u>Special Characters and Inconsistencies</u> :

- Special characters and inconsistencies might be present in text fields like DBA Name, AKA Name, Address, City, State, Violations, and others. These would require a more detailed textual analysis to identify and rectify, if necessary, for certain types of analysis or data processing.

Handling Special Characters and Inconsistencies :

● Text Normalisation: Apply text preprocessing techniques on textual fields to standardise the data. This includes converting to lowercase, removing special characters, and possibly techniques to identify similar but differently
worded violations.
● Standardisation: For fields like City and State, ensure consistency in naming (e.g., converting all to uppercase) and correct any misspellings or variations (e.g., "CHICAGO" vs. "Chicago").

5. <u>Geolocation Data</u> :

- The Latitude and Longitude columns, along with Location, provide geospatial data. The presence of missing values in these columns might limit geospatial analyses. Moreover, the accuracy and precision of this data would need to be validated for specific use cases.

Handling Geolocation Data :
● Validation and Imputation: Validate the Latitude and Longitude values to ensure they fall within expected ranges. For missing geolocation data, consider using external APIs or geocoding services to fill in missing values based on the Address.

6. <u>General Observations :</u>

- The dataset contains a wide range of information about food inspections, including the type of facility, risk level, results, and specific violations. This suggests a rich dataset that could be used

for various analyses, such as predicting inspection outcomes, identifying patterns in violations, or analysing food safety trends across different areas or facility types.

The analysis of the Dallas dataset reveals several observations and potential issues with the data:

Missing Values: Numerous columns have missing values, especially those related to violation details (e.g., Violation Points, Violation Detail, Violation Memo, etc.), which could impact analyses that rely on complete records of inspections and violations.

Handling discrepancy: Fill the void spaces with value as NULL.

Data Cleaning: As there is a separate column for location the lat long location should be separated into two different columns.

Handling discrepancy: Clean the Lat Long Location column to ensure a consistent format for geolocation data, separating latitude and longitude into separate columns if necessary for spatial analysis.

Date Formats: The Inspection Date column uses a standard date format (MM/DD/YYYY), which is suitable for analysis, but consistency and correctness need to be verified across the dataset.

Handling discrepancy: Verify the consistency of the date formats across the dataset and convert the Inspection Date column to a datetime data type for easier manipulation

Multi valued attributes and data type conversions:

Handling Multivalued attributes:

To address the multi-valued attributes in the dataset, the Violation Description fields will be concatenated into a single column by writing expressions in Talend for concatenations. Similarly, all Violation Memo fields will be concatenated into another single column. For the Violation Points, all fields will be combined into a single column, ensuring the relevant information is retained. Lastly, all Violation Detail fields will be merged into another single column. By performing these steps, the multi-valued attributes can be addressed in the dataset, condensing them into single columns while retaining the relevant information.

Handling Data Types and Conversion :

In addressing data type conversions, within the tMap component, the Zip Code column will be transformed from a string to an integer data type, similarly Inspection_ID is converted to integer datatype ensuring consistency and enabling numerical operations if required. Additionally, the Inspection Date column will be converted to the date data type, utilizing Talend's functionality to parse string dates accurately into a date format. This conversion facilitates precise handling of

dates for analysis and reporting purposes, enhancing the overall integrity and usability of the dataset within the Talend environment.

• How the schemas differ between datasets and how are you planning to merge the data

The schemas for the two datasets, "Restaurant and Food Establishment Inspections" and "Food Inspections," differ significantly, both in terms of the number of columns and the types of information they contain. Here's a breakdown of the key differences and similarities:

Differences:

1. **Detailed Violation Data**: The first dataset includes detailed violation data for up to 25 violations per inspection, including descriptions, points, details, and memos for each violation. In contrast, the second dataset condenses all violations into a single "Violations" column, which likely contains a concatenated string of all violations.

2. **Column Structure**: The first dataset has a very detailed structure that separates street address components (number, name, direction, type, unit) and includes separate columns for inspection month and year, and a "Lat Long Location" column. The second dataset, however, combines the full address into one column and does not explicitly separate the inspection date into month and year or provide a combined latitude/longitude column.

3. **Identification and Naming**: The first dataset does not appear to have a unique identifier for each inspection like the "Inspection ID" in the second dataset. Furthermore, the naming conventions for restaurants and inspection details vary between the datasets; the first dataset uses "Restaurant Name" while the second uses "DBA Name" and "AKA Name".

4. **Additional Information:** The second dataset includes columns for "License #", "Facility Type", "Risk", "City", "State", "Results", which are not explicitly present in the first dataset.

Similarities:

1. **Inspection Date and Type**: Both datasets include "Inspection Date" and "Inspection Type", which are crucial for aligning records across the two datasets.

2. **Geolocation Data**: Both datasets contain geolocation information, albeit in different formats. The first dataset offers a "Lat Long Location" column, while the second provides separate "Latitude" and "Longitude" columns.

Planning to Merge Data:

Given these differences and similarities, merging the two datasets would involve several steps:

1. **Standardizing Column Names and Formats:** Align similar columns by renaming them to match and converting data formats as necessary (e.g., splitting the "Lat_Long_Location" column in the second dataset to match the detailed address and latitude longitude structure of the first dataset).

2. **Handling Violations Data**: Since the violations are detailed extensively in the first dataset and condensed into one column in the second. This involves summarizing the detailed violations from the first dataset.
3. **Creating a Unified Schema**: Determine which columns are essential for your analysis or reporting and create a new schema that incorporates these columns. Like getting detailed violation data, geolocation information, inspection results.
4. **Data Transformation and Cleaning**: Transform data to fit the unified schema, which may involve aggregating or disaggregating data, filling in missing values, and converting data types.
5. **Merging Records**: Identifying a key or set of keys (e.g., a combination of "Inspection Date", "Inspection Type", and restaurant name) to merge records from the two datasets. Given the presence of unique identifiers in only one dataset, this step involves creating a key to align records accurately.

This approach aims to reconcile the schema differences to create a comprehensive, merged dataset that leverages the strengths and details of both sources.