talendMidterm Team Project:
Standardization and Integration of Dallas and Chicago Food Inspection Dataset

Group 1:

Prasad

Shantanu

Prarthana

Soham

Table of Contents

Project Overview:

Chicago Dataset:

This dataset has been derived from inspections of restaurants and other food establishments in Chicago from January 1, 2010 to the present. Inspections are performed by staff from the Chicago Department of Public Health's Food Protection Program using a standardized procedure. The results of the inspection are inputted into a database, then reviewed and approved by a State of Illinois Licensed Environmental Health Practitioner (LEHP).

Dallas Dataset:

This dataset conveys essential information about various establishments, including their names, physical addresses, inspection dates, overall inspection scores, and specific point deductions assigned for individual violations. It provides a comprehensive overview of the inspection process, highlighting key details such as the establishment's identity, location, inspection timeline, overall assessment, and the specific areas where point deductions occurred due to violations. The dataset serves as a valuable resource for understanding and analyzing the performance and compliance of different establishments during inspections.

Step 1: Understanding the Dataset

- Analyzed and discussed the Schema and Content of the provided dataset
- Identified key tables, fields, and relationships to establish a foundational understanding
- Noted and handled all unique characteristics or challenges within the dataset's organization
- Identified potential data quality issues, outliers, or anomalies
- Discussed a schema structure for both datasets

Step 2: Data Profiling in Alteryx

- Imported the datasets into Alteryx
- Utilized Alteryx tools for data profiling
- Further transformed the Data on Talend
- Created a dimensional model
- Loaded data in the dimensional model with Talend
- Generated visualizations to analyze the dataset for profiling process
- Noted any initial observations or patterns that might impact subsequent processing or analysis

Aim of the Assignment:

The aim of this project is to execute a comprehensive end-to-end Business Intelligence (BI) initiative. Starting with data analysis using Alteryx/Python, the project focuses on understanding data properties, documenting, creating mapping documents, staging, cleaning, and loading data into an integration schema. The final objective is to generate meaningful reports based on the integrated data, providing actionable insights for decision-makers.

Data Source Observations:

Chicago Dataset 267908

Name	Datatype	Unique values	Null values	Observation	Min Value	Max Value
Inspection_ID	V_WString	100%	0.00%	The InspectionID field, which was exclusive to the Chicago dataset and featured unique values for each record, played a crucial role in identifying individual inspection instances.	5	7
DBA Name	V_WString	11.87%	0.00%	Legal Name (DBA): The column captures the legal registration name of the food establishment with 11.87% unique values and no missing entries.	1	79
AKA_Name	V_WString	11.32%	0.92%	This field records the name by which the establishment is commonly recognized, exhibiting 11.32% uniqueness and minimal missing data (0.92%).	2	79
License	V_WString	16.61%	0.01%	An identifier assigned by the Department of Business Affairs and Consumer Protection for licensing purposes, with 16.61% uniqueness and negligible missing entries (0.01%).	1	7
Facility_Type	V_WString	0.17%	1.91%	Classifies establishments into various categories such as bakeries, grocery stores, and restaurants, with 0.17% uniqueness and 1.91% null values.	3	47
Inspection_Type	V_WString	0.04%	0.00%	Categorizes various inspection types such as routine canvass and complaint responses, with very low uniqueness (0.04%) and no missing data.	3	41
Inspection_Date	V_WString	1.33%	0.00%	Records the date of each inspection, showing 1.33% uniqueness and no missing entries.	10	10

Risk	V_WString	0.001%	0.03%	Establishments are categorized into risk levels, with Category 1 representing the highest risk; this column has a minimal uniqueness of 0.001% and 0.03% missing entries.	1	88
Address	V_WString	7.27%	0%	Captures complete address details of street address, exhibiting 7.27% uniqueness and no missing data.	9	51
City	V_WString	0.03%	0.06%	Location details of the establishment with 0.03% unique values and 0.06% null values	2	20
State	V_WString	0.001	0.02	Location details of the establishment with 0.01% unique values and 0.02% null values	2	2
Zip	V_WString	0.05%	0.02%	Location details of the establishment with 0.05% unique values and 0.02% null values	5	5
Results	V_WString	0.02%	0%	This column records the outcomes of inspections, indicating whether the establishments passed, passed with conditions, or failed, with minimal missing data (0.02%).	4	20
Violation	V_WString	72%	27.42%	This column lists all violations detected at the facility in a pipe-separated format, exhibiting significant uniqueness and a substantial number of missing entries (27.42%)	30	254
Latitude	V_WString	6.8%	0.35%	Provides latitude coordinates with 6.8% uniqueness and 0.35% missing data.	12	18
Longitude	V_WString	6.8%	0.35%	Records longitude coordinates with 6.8% uniqueness and 0.35% missing data.	13	18
Location	V_WString	6.8%	0.35%	'Location' field exhibit moderate uniqueness and minimal missing data, facilitating precise mapping of inspection sites.	34	40

Dallas Dataset:

Name	Datatype	Unique values	Null values	Observation	Min Value	Max Value
Restaurant Name	V_WString	11.65%	0.01%	Records the names of food establishments undergoing inspection, with 11.65% uniqueness and minimal null values (0.01%).	3	65
Inspection Type	V_WString	0.003%	0%	Describes the nature of the inspection process, with very low uniqueness (0.003%) and no null values.	7	9
Inspection Date	V_WString	2.92%	0%	Captures the dates of inspections conducted, exhibiting moderate uniqueness (2.92%) and no null values.	10	10
Inspection Score	V_WString	0.07%	0%	Represents the numerical scores assigned to establishments based on inspection findings, with a low uniqueness of 0.07% and no null values.	1	3
Street Number	V_WString	4.39%	0	These fields collectively provide detailed address information, with moderate to high uniqueness and very few null values.	1	5
Street Name	Date	1.07%	0%		2	25
Street Direction	V_WString	0.01%	66.98%		1	1
Street Type	V_WString	0.02%	2.12%		2	4
Street Unit	V_WString	1.26%	64.36%		1	5
Street Address	V_WString	9.97%	0%	A concatenated field comprising street number, name, direction, and type, with high uniqueness (9.97%) and no null values.	10	37
Zip Code	V_WString	0.20%	0%	Records the ZIP codes of establishments, showing moderate uniqueness	5	10

				(0.20%) and no null values.		
Violation Description	V_WString	-	-	This column provides detailed descriptions of violations detected during inspections, they range from 1-25.	-	-
Violation Points	V_WString	-	-	This column denotes the numerical points assigned to each violation detected during inspections	-	-
Violation Detail	V_WString	-	-	Field used to describe the type of violation associated with the enforcement action.	-	-
Violation Memo	V_WString	-	-	This field is used for any additional comments about the enforcement action.	-	-
Inspection Month	V_WString	0.11%	0%	Records the month in which inspections were conducted, exhibiting a low uniqueness of 0.11% and no null values.	8	8
Inspection Year	V_WString	0.01%	0%	Captures the year of inspections, with very low uniqueness (0.01%) and no null values.	6	6
Lat Long Location	V_WString	24.6%	0%	Represents the combined latitude and longitude coordinates of establishments, with moderate uniqueness (24.6%) and no null values.	10	73

In summary, the provided table contains a mix of columns with varying characteristics:

- Legal Name (DBA): The legal registration name of the food establishment.
- Public Name (AKA): The name by which the establishment is commonly recognized.
- License Number: A unique identifier assigned by the Department of Business Affairs and Consumer Protection for licensing purposes.
- Facility Type: Classifications include a diverse range of establishments such as bakeries, grocery stores, restaurants, schools, and more, each with specific descriptors.

- Risk Category: Establishments are categorized into three levels based on the potential health risks they pose, with Category 1 being the highest risk. Inspection frequency correlates with the risk level, with higher risk facilities inspected more frequently.
- Location Information: This includes the complete address of the establishment (street address, city, state, and ZIP code).
- Inspection Date: The specific date on which an inspection took place. Establishments typically undergo multiple inspections, each recorded separately.
- Inspection Type: Various types of inspections are conducted, including routine canvass, pre-opening consultations, complaint responses, license issuance, and task-force inspections, among others. Re-inspections are noted accordingly.
- Inspection Results: Outcomes are categorized as 'pass', 'pass with conditions', or 'fail', based on the presence and correction of critical or serious violations at the time of inspection.
- Violations: The dataset documents up to 45 distinct violations, with each record detailing the requirement for non-violation and the specific findings that led to any violations being issued.

The dataset contains essential attributes such as 'DBA' (Doing Business As), 'AKA' (Also Known As), 'License Number', 'Type of Facility', 'Risk Category', and full address details, which exhibit low to moderate uniqueness and are completely filled, indicating robust data collection without missing entries. Fields like 'Inspection Date' and 'Type' show moderate uniqueness and very low missing data, ensuring a comprehensive timeline and categorization of inspections. The 'Results' field, crucial for understanding inspection outcomes, and the detailed 'Violations' listings are meticulously recorded, though the latter might require normalization to ensure consistency across entries. The 'Risk Category' of facilities, pivotal for prioritizing inspections based on potential health impact, along with the 'Type of Facility', provides a nuanced view of the diverse food establishments within the dataset. Inspection outcomes ('Results') are categorically noted, offering insights into compliance levels, with 'Violations' detailing specific issues found, crucial for targeted improvements.This dataset is devoid of significant gaps in critical fields, suggesting a high level of data integrity and reliability for analysis. However, given the detailed nature of 'Violations', some level of data cleansing might be beneficial to address any inconsistencies or redundancies, enhancing the dataset's utility for monitoring and improving food safety standards in Chicago.

The Dallas food inspection dataset, covering inspections from October 2016 to the present, contains a comprehensive set of fields that detail the inspection activities of restaurants and food establishments in Dallas. Here are the key data elements from this dataset:

- Restaurant Name: The name of the food establishment being inspected.
- Inspection Type: The nature of the inspection, typically noted as "Routine".
- Inspection Date: The date on which the inspection was conducted.
- Inspection Score: A numerical score assigned to the establishment based on the inspection findings.
- Street Number, Street Name, Street Direction, Street Type, Street Unit: These fields together provide the full address of the food establishment.
- Street Address: A concatenation of the street number, name, direction, and type, providing a complete address in a single field.
- City, State, Zip Code: Location details of the establishment.
- Violation Descriptions and Violation Points: For each inspection, there can be multiple violations described, each accompanied by points that likely contribute to the overall inspection score. These are split across numerous columns, each denoting a specific violation and its details.
- Inspection Month and Inspection Year: These fields provide temporal context to the inspection, categorizing it within a specific month and fiscal year.
- Lat Long Location: This field contains both the street address and the geographical coordinates (latitude and longitude) of the establishment, useful for mapping and spatial analysis.

This dataset is structured to provide a detailed account of each inspection, including the outcomes and specific areas of non-compliance. The inclusion of violation details alongside scores and precise location data makes it a valuable resource for analyzing food safety and compliance trends within Dallas.

Key observations from the Dallas food inspection dataset, covering inspections from October 2016 to the present, can be summarized as follows:

The dataset meticulously captures a wide array of attributes, including the 'Restaurant Name', 'Inspection Type', 'Inspection Date', and 'Inspection Score', indicating a thorough and systematic approach to data collection. These fields are consistently populated, demonstrating a high degree of completeness and a low incidence of missing data. The 'Inspection Type' and 'Date' fields, in particular, offer a detailed chronology and categorization of inspections, showcasing moderate uniqueness and precision in tracking inspection activities over time.

Address details are extensively documented through fields such as 'Street Number', 'Name', 'Direction', 'Type', and 'Unit', along with 'City', 'State', and 'Zip Code', ensuring accurate localization of establishments. This granular address information, coupled with the 'Lat Long Location' that combines textual and geographical data, underscores the dataset's capacity for detailed spatial analysis and mapping of food safety inspections across Dallas.

Violation-related information is captured in a series of columns detailing specific violations and their points, reflecting a nuanced approach to documenting compliance issues. Despite the structured presentation, the wide range of violations necessitates normalization and careful handling to ensure clarity and consistency across records.

Temporal aspects of the data are well-represented through the inclusion of 'Inspection Month' and 'Year', facilitating trend analysis and the assessment of inspection outcomes over time. This temporal granularity supports the identification of patterns and potential seasonal or annual shifts in inspection findings or establishment compliance.

Overall, the dataset demonstrates robust data integrity and reliability, with comprehensive coverage of essential attributes relevant to food establishment inspections. While the detailed nature of violation data presents some challenges in terms of standardization and consistency, these are outweighed by the dataset's strengths in facilitating a thorough analysis of food safety and public health trends in Dallas.

SCHEMA EXAMINATION:

The schema structure was chosen based on a thorough examination and integration of the food inspection datasets from both Chicago and Dallas. Here are some of the reasons and descriptions for the schema structure, as well as the quality of the schema based on the analysis of the provided document:

Reasons for Schema Structure:

1. Uniform Record Identification:
   - A unique `InspectionID` was introduced to the Dallas dataset to align with the Chicago dataset, ensuring each record can be identified consistently across both datasets.

2. Data Integrity and Integration:
   - Columns like `License`, `Facility_Type`, and `State` were adjusted or added where necessary to align both datasets, allowing for seamless integration and maintaining data integrity.

3. Temporal Analysis Support:
   - `Inspection Date` was maintained to facilitate trend analysis and identify seasonal or annual patterns in inspection outcomes.

4. Comprehensive Coverage:
   - Essential attributes for food establishment inspections were covered comprehensively, including `Facility Name`, `Inspection Type`, `Risk`, and `Violation Details`.

5. Consistency in Evaluation:
   - For `Inspection Score` and `Risk`, consistent evaluation methods were applied to both datasets. The Chicago dataset had its Inspection Score calculated, while the Dallas dataset's Risk was classified based on score distribution.

Schema Structure Description:

- InspectionID: Unique identifier for each record.
- Facility_Name: Name of the establishment.
- License: Licensing information.
- Facility_Type: Type of establishment, with a newly introduced column in the Dallas dataset.
- Inspection Type: Nature of the inspection.
- Inspection Date: Date when the inspection occurred.
- Inspection Score: Numerical score based on findings, standardized across datasets.
- Risk: Risk level assigned to the establishment based on score distribution analysis.
- Address Information: Detailed address, city, state, and zip code for location accuracy.
- Violation Description: Specific violations observed during the inspection and their associated points.

Quality of the Schema:

- Robustness: The schema is robust, with a comprehensive set of attributes essential for a thorough analysis of food safety and compliance.
- Data Integrity: The schema maintains high data integrity with systematic record identification and comprehensive coverage of attributes.
- Consistency: There is consistency in data representation, with standardized column names and data types across both datasets.
- Analytical Capability: The schema supports analytical capabilities such as trend analysis, spatial analysis, and compliance evaluation.

The final schema is considered good due to its comprehensive nature, ability to integrate diverse datasets, and support for a wide range of analytical needs. The detailed nature of the violation data and the inclusion of temporal aspects allow for nuanced analysis, although it may present challenges in standardization and consistency which are outweighed by the schema's strengths.

In conclusion, the schema structure for the dataset integration according to us is well-justified and constructed with careful consideration of data integrity, consistency, and analytical utility. It demonstrates a high level of thoughtfulness in its ability to handle complex and detailed information effectively, making it a solid foundation for subsequent data analysis and reporting.

COLUMN WISE ANALYSIS OF KEY TRANSFORMATIONS AND DECISIONS(How the schemas differ between datasets and how are you planning to merge the data):

-InspectionID: - Unique `InspectionID` values, originally present only in the Chicago dataset, were systematically generated and added to the Dallas dataset to ensure uniform record identification. This step was crucial for maintaining data integrity and facilitating the seamless merging of both datasets into a single, structured table. By standardizing the identifier across datasets, we ensured precise tracking and analysis of inspections from both cities in the consolidated dataset.

-Facility_Name: The name of the food establishment being inspected, both the datasets had this information and no transformation was carried out for this column

-License: The License column, present in the Chicago dataset but absent from the Dallas dataset, necessitated schema alignment to maintain a common structure across both datasets. To address this, we added a License column in the Dallas dataset, populated with null values. This adjustment was critical for preserving a uniform schema, facilitating data integration and ensuring consistency in the merged dataset's schema architecture.

-Facility_Type: The Facility_Type column, available in the Chicago dataset but absent in the Dallas dataset, required alignment for consistency. In the Dallas dataset, establishments were indicated by the Restaurant_Name column, implying a primary focus on restaurants. To bridge this structural difference, we introduced a Facility_Type column in the Dallas dataset, assigning all entries the value 'Restaurant' to reflect the nature of the establishments as inferred from the original column naming.

-Inspection Type: Inspection Type Uniformity: The Inspection Type column, found in both the Chicago and Dallas datasets, included both diverse inspection categories conducted across establishments. This column's presence and consistency in both datasets ensured alignment in inspection categorization methodology, facilitating seamless analysis and comparison of inspection types between the two datasets.

-Inspection Date: The Inspection Date column, featured in both the Chicago and Dallas datasets, denotes the precise date of each inspection occurrence. Given that establishments undergo multiple inspections, each documented individually, the presence of this column in both datasets ensures standardized recording of inspection dates across the board. This consistency enables comprehensive analysis and comparison of inspection timelines between the two datasets.¬

-Inspection Score: Inspection Score Calculation: In the Dallas dataset, the Inspection Score column was already present. For the Chicago dataset, the Inspection Score was calculated as the sum of Risk Score, Result Score, and Sum Violation Point. Any score that exceeded 100 was capped at 100 to maintain consistency and standardization. This calculation method ensures uniformity in assessing inspection performance across both datasets.

-Risk: While the Inspection Score column was already provided in the Chicago dataset, for the Dallas dataset, we computed it by analyzing the distribution of Inspection Scores. We observed that the majority of scores fell within the range of 54 to 99. To categorize these scores, we segmented them into brackets of 15 points each. Subsequently, we classified the scores as low, medium, or high based on their distribution within these brackets. This approach ensures consistency in evaluating inspection performance across both datasets by categorizing scores according to their relative frequency and distribution.

-Violation Number: The original dataset for both Chicago and Dallas featured multiple violations corresponding to each Inspection ID. While Chicago had them in the same column, Dallas had them separated by columns. To restructure the data for improved analysis, we transposed these columns, grouping them by Inspection ID and converting them into rows. For Chicago the Violations were pipe separated and hence we used ‘Text to Columns’ to separate the data into columns first and then transposed these columns, grouping them by Inspection ID and converting them into rows. To accurately track the number of violations associated with each Inspection ID, we introduced a new column named "violation number." This column assigns an index to each violation count within its respective inspection row, facilitating granular tracking and analysis of violations across inspections.

-Violation_Category_ID & Violation_Description: In the Chicago dataset, Violation Category IDs and Violation Descriptions were derived from the Violations column by adding a delimiter of a period ('.'). Similarly, in the Dallas dataset, a ‘Violation Description’ column was created by parsing the Violations column, also using a delimiter to separate the definitions. The Violation Category ID contains all the numerical values preceding the descriptions, while the Violation Description column includes the detailed definitions extracted from both the Violations and Violation Description columns, ensuring consistency in data representation and schema structure.

-Violation Point: In the Dallas dataset, Violation Points were provided as a column adjacent to each violation description. However, for the Chicago dataset where violations were uniformly advised with 2 points, a new column was introduced to represent Violation Points, assigning a value of 2 to all violations. This approach ensures consistency in the representation of violation severity across both datasets.

-Sum Violation Point: A new aggregated column, "Sum Violation Point," was introduced in both datasets to represent the total sum of violation points grouped by InspectionID. This column provides insight into the cumulative violation points for each inspection instance, aiding in the assessment of overall inspection severity.

-Address: In the initial datasets of both Chicago and Dallas, address details were stored in columns denoted as "Street Address" and "Address," encapsulating the complete street address within a unified field for each entry. This consolidation of address information into a single field streamlines data representation and enhances ease of access and analysis.

-City: In the Chicago dataset, entries included various cities beyond Chicago itself. To ensure data consistency and accuracy, a validation process was implemented by cross-referencing the city name with the corresponding zip code using a comprehensive zip code dataset in Talend. This validation helped identify and standardize the actual city names associated with each entry. Conversely, in the Dallas dataset, all entries originated from Dallas, negating the need for such validation. However, for schema uniformity and structural alignment, a new column was introduced in the Dallas dataset, with all values set to "Dallas." This standardization approach enhances dataset coherence and facilitates seamless integration and analysis.

-State: In the Chicago dataset, state names were assigned to each entry based on the corresponding cities using a VLOOKUP. This ensured that each record was accurately associated with its respective state. Conversely, in the Dallas dataset, all entries originated from Dallas, thus implicitly indicating the state as Texas (TX). To maintain schema consistency and structural coherence, a state column was introduced in the Dallas dataset, with all values uniformly set to "TX." This standardization ensures a unified representation of state information across both datasets, enhancing data integrity and facilitating seamless analysis and integration.

-Zip Code: Both the Chicago and Dallas datasets featured a column labeled "ZipCode" and “Zip,” providing precise location information crucial for standardizing and validating city column entries. This shared attribute facilitated cross-referencing and validation processes, ensuring accuracy in city data and enhancing overall dataset integrity.

-Latitude: In the Dallas dataset, latitude and longitude values were combined into a single column labeled 'Lat Long Location,' while the Chicago dataset maintained them separately, with latitude in its own column. To align with the expected schema structure and facilitate analysis, we utilized the 'Text to Columns' tool, setting the delimiter as a comma (',') to split the combined values. This extraction process separated latitude into a distinct column, ensuring consistency across datasets and adhering to the desired schema format.

-Longitude: In the Dallas dataset, longitude and latitude values were combined into a single column labeled 'Lat Long Location,' while the Chicago dataset maintained them separately, with longitude in its own column. As mentioned before, to align with the expected schema structure and facilitate analysis, we utilized the 'Text to Columns' tool, setting the delimiter as a comma (',') to split the combined values. This extraction process separated longitude into a distinct column, ensuring consistency across datasets and adhering to the desired schema format.

Data Quality Assessment:

Converting Data Types:

The transformation of data types for certain columns in both datasets serves to enhance data consistency, accuracy, and analytical capabilities.

1. License Number: Converting License Number to an integer type (int64) streamlines data storage and processing, as license numbers typically consist of numeric values. This transformation facilitates efficient numerical operations and comparisons when analyzing licensing information across establishments.

2. Inspection Score: By converting Inspection Score to an integer type (int64), numerical calculations and comparisons become more straightforward and efficient. This change allows for easier aggregation, statistical analysis, and visualization of inspection scores, aiding in the evaluation of compliance levels and establishment performance.

3. Violation Point and Sum Violation Point: Transforming Violation Point and Sum Violation Point to integer types (int64) enables precise numerical representation of violation severity and aggregation of violation points for each inspection instance. This conversion facilitates comprehensive analysis of violation trends, prioritization of corrective actions, and evaluation of overall inspection outcomes.

4. Inspection Date: Converting Inspection Date to a Date type ensures standardized handling of date values, facilitating chronological sorting, filtering, and analysis of inspection data. This transformation enhances the accuracy and reliability of temporal analyses, enabling insights into inspection trends over time.

Overall, these data type conversions optimize data management and analysis processes, promoting consistency, efficiency, and accuracy in exploring and interpreting inspection datasets from both Chicago and Dallas.

Column Dropping Rationale:
Dallas Dataset:

1. Street Number, Street Name, Street Direction, Street Type, Street Unit: These columns were concatenated and added to the Street Address column. Maintaining redundant information in separate columns became unnecessary and inefficient for data storage and processing. Given that the concatenated Street Address column already encapsulates complete address details, retaining individual components would result in data redundancy and increased complexity without adding significant value to analysis or interpretation.

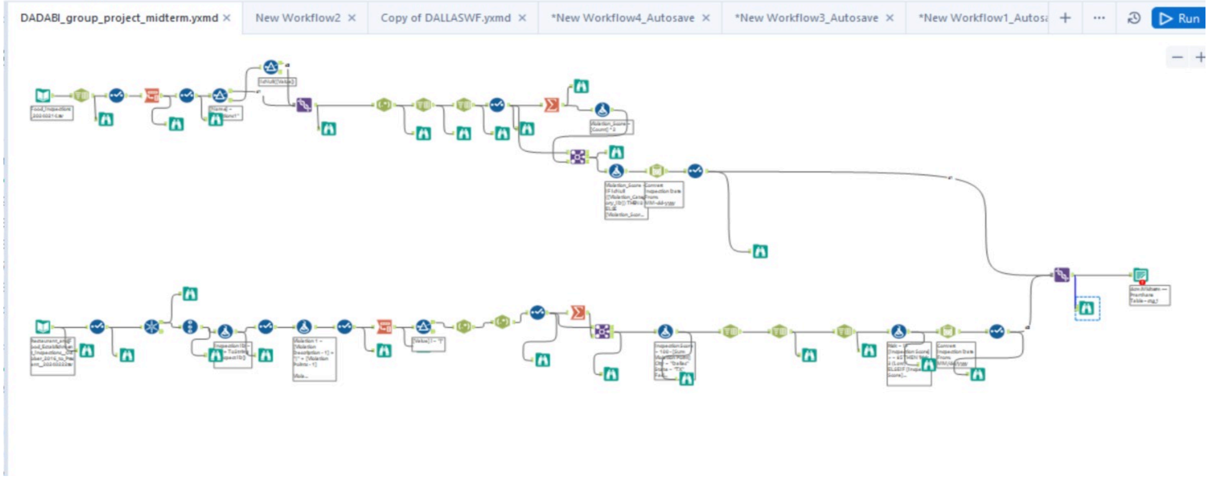
2. Violation Detail and Violation Memo: These columns, intended to provide additional context or descriptions for violations, did not offer concise or informative explanations of the violations detected during inspections. Instead, they functioned more as metadata for the Violation Description column. Omitting these columns streamlines the dataset by removing redundant information and focusing on the essential details needed for analysis. This approach enhances dataset clarity and interpretability, ensuring that the data remains relevant and actionable for stakeholders without unnecessary verbosity or redundancy.

Chicago Dataset:

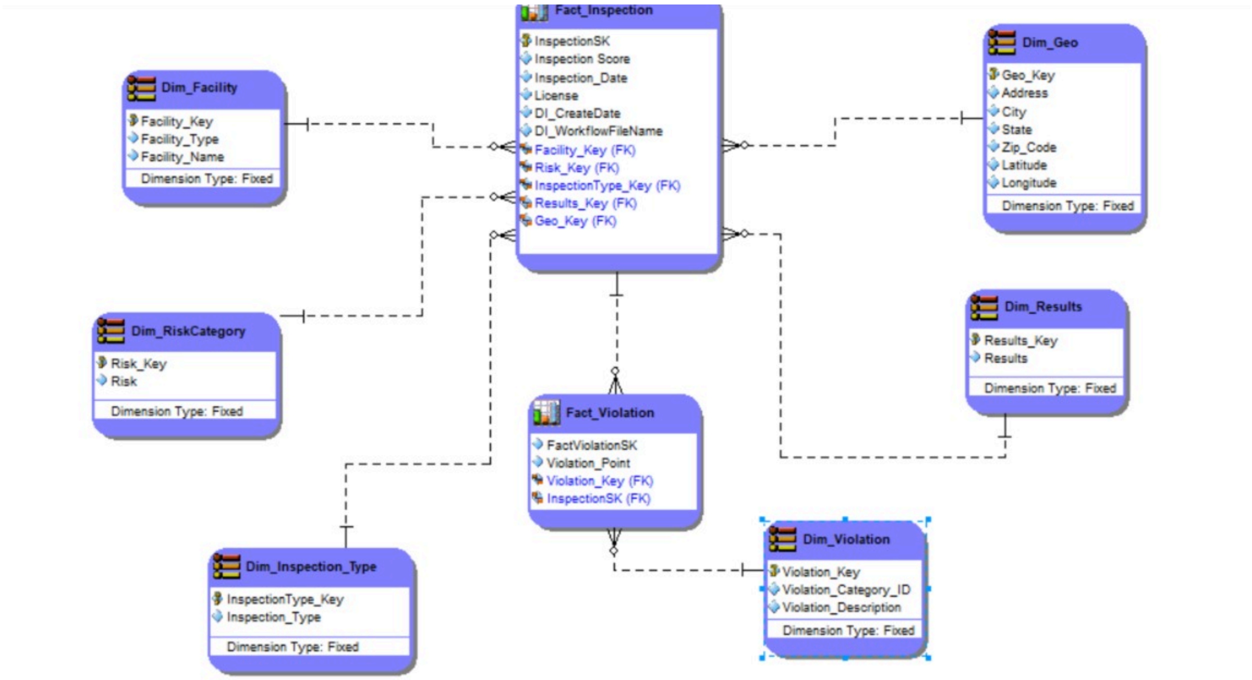
1. AKA Name: The AKA Name column, which stands for "Also Known As," provided alternative names for the Restaurant Name but exhibited similarity to the primary Restaurant Name column. Given their comparable nature and limited additional value for analysis or stakeholder insights, maintaining both columns became redundant and unnecessarily complex. Therefore, the AKA Name column was dropped to streamline the dataset and avoid redundancy in data representation.

2. Location: The Location column, comprised of concatenated Latitude and Longitude coordinates, essentially duplicates information already present in separate Latitude and Longitude columns. Retaining the Location column adds unnecessary redundancy to the dataset without offering any additional analytical value. Therefore, dropping the Location column simplifies the dataset structure and ensures a more efficient use of resources for data storage and processing.

Alteryx Profiling workflow screenshot:


DADABI_group_project_midterm.yxmd ×
New Workflow2 ×
Copy of DALLASWF.yxmd ×
*New Workflow4_Autosave ×
*New Workflow3_Autosave ×
*New Workflow1_Autosa
Run

3. **Dimensional Data Modeling (Part 2)**
    - ER/Studio Dimensional Model Screenshot



The dimensional model is a star schema composed of dimension tables and fact tables.
Dimension Tables:

1. Dim_Facility
   - Primary Key: `Facility_Key`
   - Attributes: `Facility_Name`, `Facility_Type`
   - This dimension table holds descriptive information about facilities, such as the name and type. The primary key, `Facility_Key`, uniquely identifies each facility.

2. Dim_Geo
   - Primary Key: `Geo_Key`
   - Attributes: `Address`, `City`, `State`, `Zip_Code`, `Latitude`, `Longitude`
   - The geographic dimension table contains location-related information for an entity such as an inspection site. It includes detailed address information and geographical coordinates.

3. Dim_RiskCategory
   - Primary Key: `Risk_Key`
   - Attributes: `Risk`
   - This dimension table categorizes the risk levels, which could be associated with facilities or inspection results.

4. Dim_Results
   - Primary Key: `Results_Key`
   - Attributes: `Results`
   - The results dimension table captures the potential outcomes of inspections.

5. Dim_Inspection_Type
   - Primary Key: `InspectionType_Key`
   - Attributes: `Inspection_Type`
   - This dimension table describes types of inspections that can occur.

6. Dim_Violation
   - Primary Key: `Violation_Key`
   - Attributes: `Violation_Category_ID`, `Violation_Description`

  - This table contains information about various types of violations that can be recorded during inspections.

Fact Tables:

1. Fact_Inspection
  - Primary Key: `InspectionSK`
  - Attributes: `Inspection_Score`, `Inspection_Date`, `License`, `DL_CreateDate`, `DL_WorkflowFilename`
  - Foreign Keys:
    - `Facility_Key (FK)` references `Dim_Facility`
    - `Risk_Key (FK)` references `Dim_RiskCategory`
    - `InspectionType_Key (FK)` references `Dim_Inspection_Type`
    - `Results_Key (FK)` references `Dim_Results`
    - `Geo_Key (FK)` references `Dim_Geo`
  - The fact table for inspections contains quantitative data about each inspection event, such as scores and dates, and references to related dimensions for contextual analysis.

2. Fact_Violation
  - Primary Key: `FactViolationSK`
  - Attributes: `Violation_Point`
  - Foreign Keys:
    - `Violation_Key (FK)` references `Dim_Violation`
    - `InspectionSK (FK)` references `Fact_Inspection`
  - This fact table records specific violations, noting the points associated with each violation and linking back to the inspection it was recorded during.

Relationships and Analysis:
- One-to-Many: Each foreign key in the fact table corresponds to a primary key in a dimension table, forming a one-to-many relationship (one dimension record relates to many fact records). This is typical of star schemas where dimensions provide descriptive context for the numerical measures in the fact tables.

The keys serve as the backbone of the data warehouse's relational structure, enabling efficient querying and reporting capabilities. When data is loaded into this model, the FKs in the fact tables will be populated with the corresponding PK values from the dimension tables, maintaining referential integrity and allowing for rich, multi-dimensional reporting.

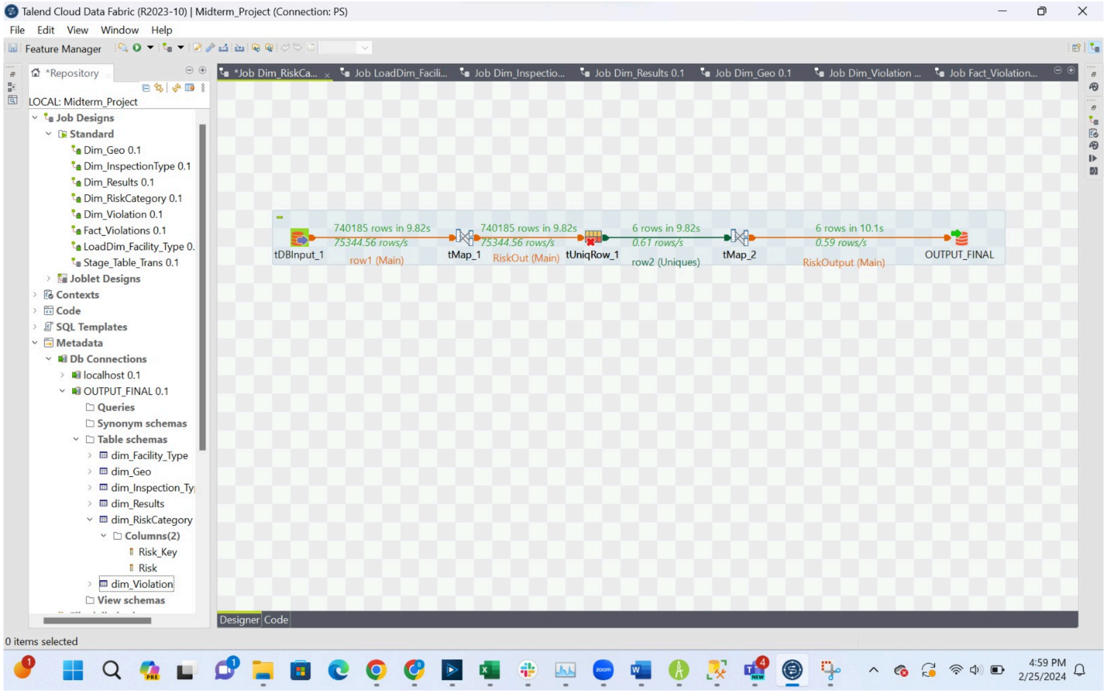
SQL SCRIPTS:

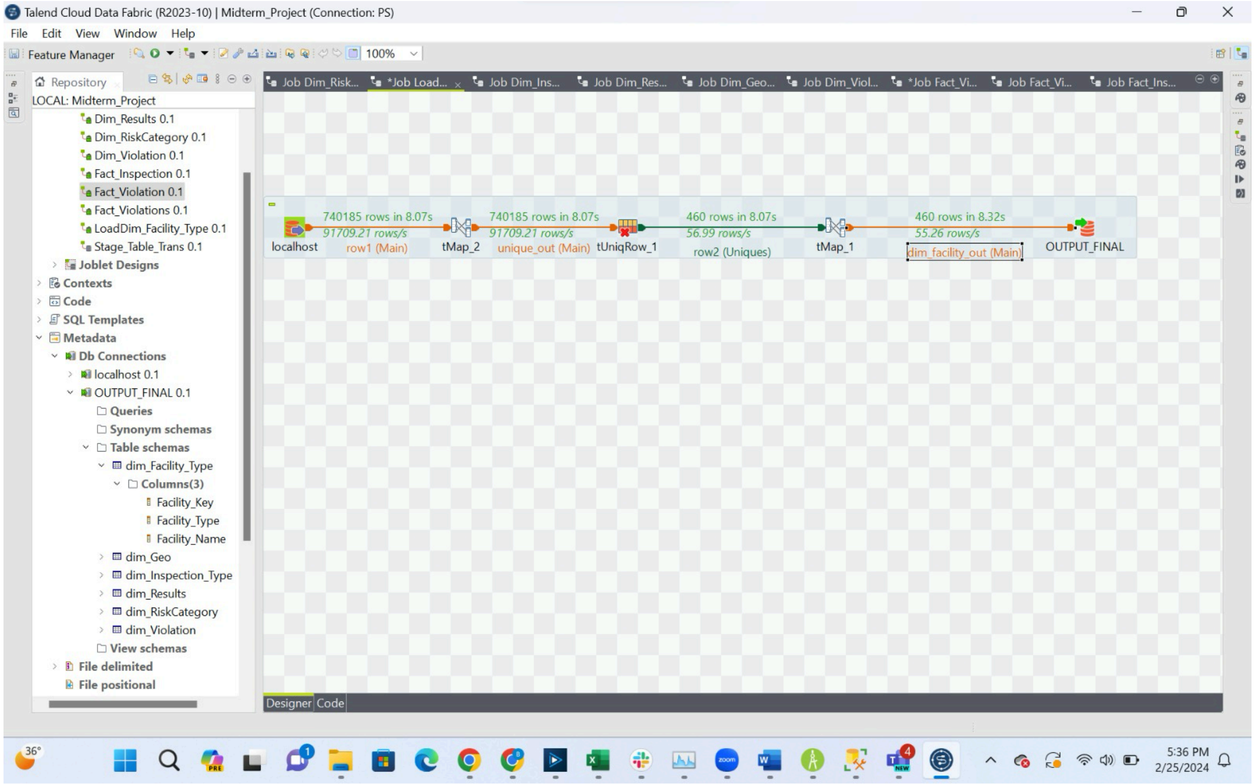
```
USE [Midterm]
GO
/** Object:  Table [dbo].[stg_2]    Script Date: 2/25/2024 8:13:28 AM **/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[stg_2](
	[Inspection_SK] [int] IDENTITY(1,1) NOT NULL,
	[Inspection ID] [nvarchar](1000) NULL,
	[Facility Name] [nvarchar](1000) NULL,
	[License] [bigint] NULL,
	[Facility Type] [nvarchar](1000) NULL,
	[Inspection Type] [nvarchar](1000) NULL,
	[Inspection_Date] [date] NULL,
	[Inspection Score] [bigint] NULL,
	[Violation Number] [char](1000) NULL,
	[Violation_Category_ID] [nvarchar](1000) NULL,
	[Violation_Description] [nvarchar](1000) NULL,
	[Violation Point] [bigint] NULL,
	[Sum Violation Point] [bigint] NULL,
        [Risk] [nvarchar](1000) NULL,
        [Address] [nvarchar](1000) NULL,
        [City] [varchar](1000) NULL,
        [State] [nvarchar](1000) NULL,
        [Zip Code] [nvarchar](1000) NULL,
        [Results] [nvarchar](1000) NULL,
        [Latitude] [nvarchar](1000) NULL,
        [Longitude] [nvarchar](1000) NULL,
        [DI_CreateDate] [datetime] NULL,
        [DI_WorkflowFileName] [varchar](1000) NULL
) ON [PRIMARY]
GO


-- Creating the Dim_Facility_Type table
CREATE TABLE Dim_Facility (
   FacilitySK INT PRIMARY KEY IDENTITY(1,1), -- Assuming surrogate key with auto-increment
   Facility_Type VARCHAR(1000),
        Facility_Name VARCHAR(1000),
        [Address] VARCHAR(1000),
        City VARCHAR(1000),
        Zip_Code VARCHAR(1000)
);

select * from dim_Inspection_Type;

select * from dim_Results;

ALTER TABLE Dim_Facility
--ALTER COLUMN Facility_Type VARCHAR(1000)
--ALTER COLUMN Facility_Name VARCHAR(1000)
--ALTER COLUMN [Address] VARCHAR(1000)
--ALTER COLUMN City VARCHAR(1000)
ALTER COLUMN Zip_Code VARCHAR(1000)

create database target;




select * from Dim_Facility
-- Creating the Dim_RiskCategory table
CREATE TABLE Dim_RiskCategory (
   RiskSK INT PRIMARY KEY IDENTITY(1,1),
   Risk NVARCHAR(255)
);

-- Creating the Dim_Geo table
CREATE TABLE Dim_Geo (
   GeoSK INT PRIMARY KEY IDENTITY(1,1),
   Address NVARCHAR(255),
   City NVARCHAR(255),
   State NVARCHAR(255),
   ZipCode NVARCHAR(20),
   Latitude FLOAT,
   Longitude FLOAT
```

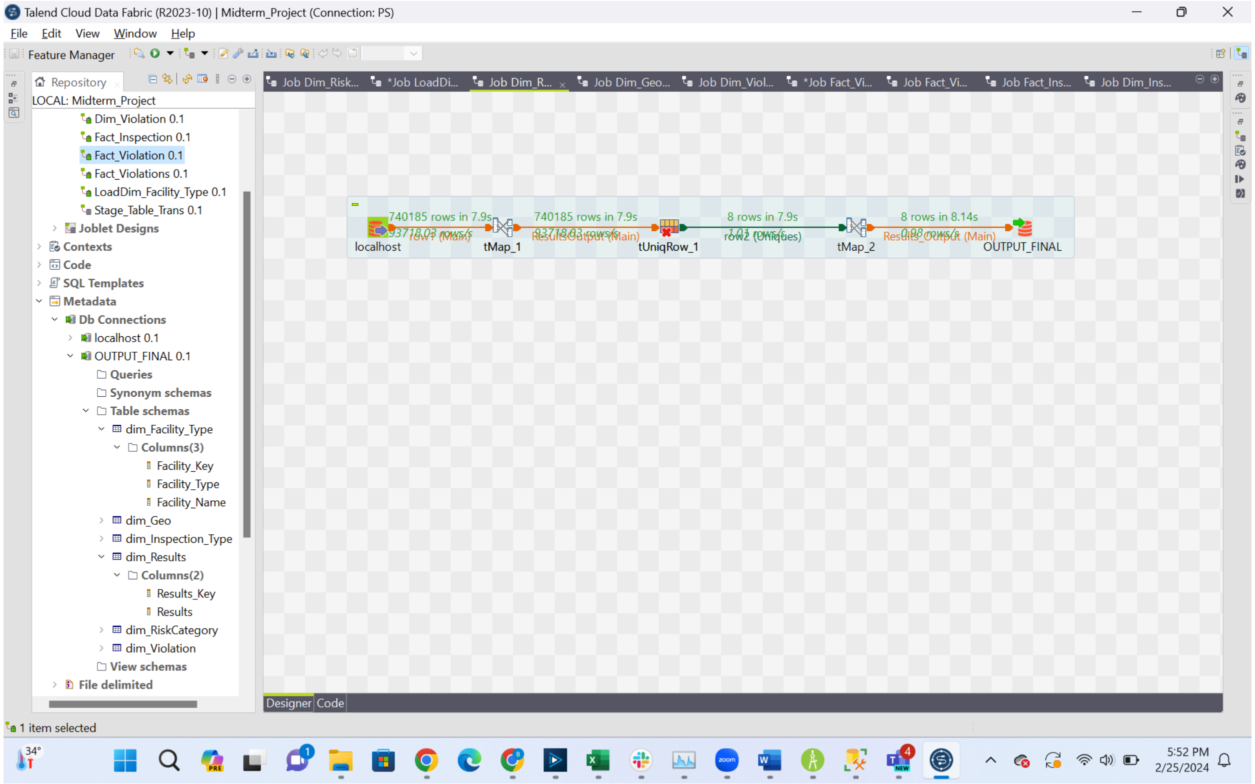
```
);

-- Creating the Dim_Restaurant table
CREATE TABLE Dim_Restaurant (
   RestaurantSK INT PRIMARY KEY IDENTITY(1,1),
   FacilityName NVARCHAR(255),
   Address NVARCHAR(255),
   ZipCode NVARCHAR(20)
);

-- Creating the Dim_Inspection_Type table
CREATE TABLE Dim_Inspection_Type (
   Inspection_TypeSK INT PRIMARY KEY IDENTITY(1,1),
   InspectionType NVARCHAR(255)
);

-- Creating the Dim_Results table
CREATE TABLE Dim_Results (
   ResultsSK INT PRIMARY KEY IDENTITY(1,1),
   Results NVARCHAR(255)
);

-- Creating the Dim_Violation table
CREATE TABLE Dim_Violation (
   ViolationSK INT PRIMARY KEY IDENTITY(1,1),
   Violation_Category_ID INT,
   Violation_Description NVARCHAR(255),
   "Version" NVARCHAR(50)
);

-- Creating the Fact_Inspection table
CREATE TABLE Fact_Inspection (
   InspectionSK INT PRIMARY KEY IDENTITY(1,1),
   InspectionScore DECIMAL(5,2),
   Inspection_Date DATETIME,
   License NVARCHAR(50),
   DL_CreateDate DATETIME,
   DL_WorkflowFileName NVARCHAR(255),
   FacilitySK INT FOREIGN KEY REFERENCES Dim_Facility_Type(FacilitySK),
   RiskSK INT FOREIGN KEY REFERENCES Dim_RiskCategory(RiskSK),
   Inspection_TypeSK INT FOREIGN KEY REFERENCES Dim_Inspection_Type(Inspection_TypeSK),
   RestaurantSK INT FOREIGN KEY REFERENCES Dim_Restaurant(RestaurantSK),
   GeoSK INT FOREIGN KEY REFERENCES Dim_Geo(GeoSK),
   ResultsSK INT FOREIGN KEY REFERENCES Dim_Results(ResultsSK)
);

-- Creating the Fact_Violation table
CREATE TABLE Fact_Violation (
   FactViolationSK INT PRIMARY KEY IDENTITY(1,1),
   Violation_Point INT,
   ViolationSK INT FOREIGN KEY REFERENCES Dim_Violation(ViolationSK),
   InspectionSK INT FOREIGN KEY REFERENCES Fact_Inspection(InspectionSK) -- Assuming that this should reference
Fact_Inspection
);
```

4. Data Loading (Part 3)

Talend Workflow







Talend Cloud Data Fabric (R2023-10) | Midterm_Project (Connection: PS)

LOCAL: Midterm_Project

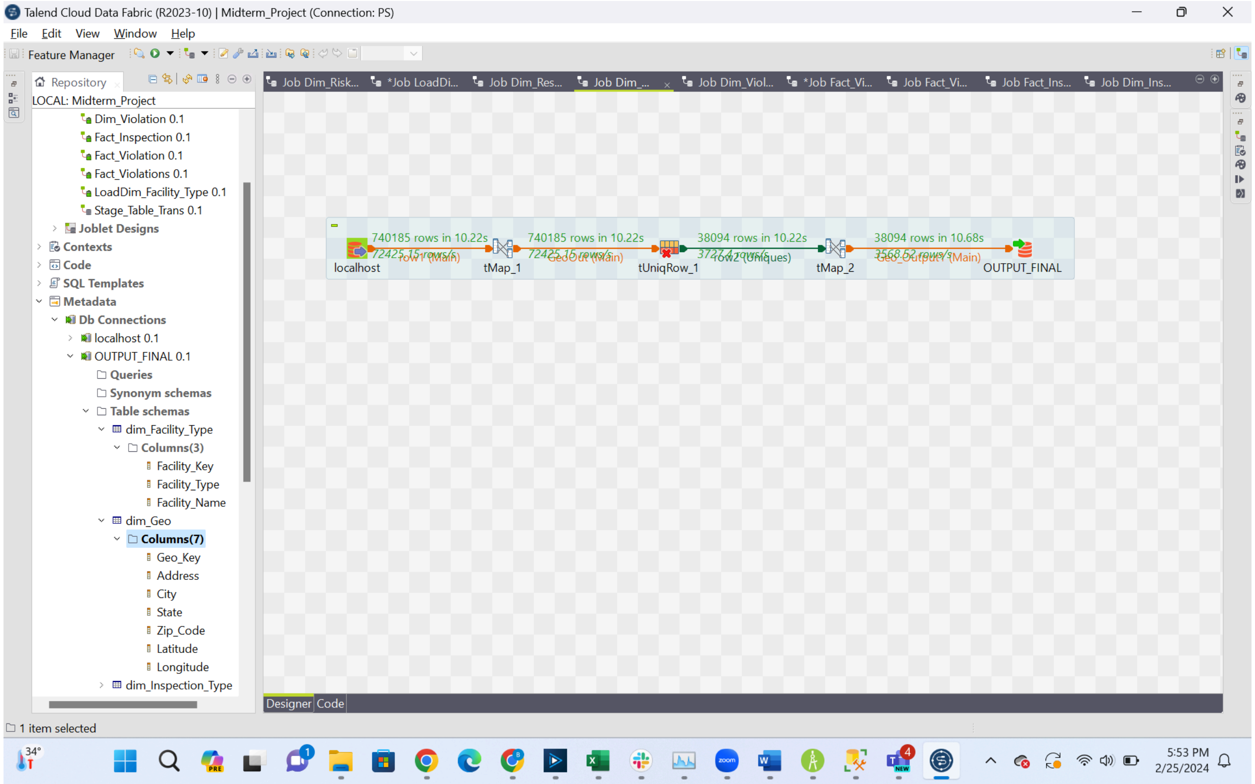
- Dim_Violation 0.1
- Fact_Inspection 0.1
- Fact_Violation 0.1
- Fact_Violations 0.1
- LoadDim_Facility_Type 0.1
- Stage_Table_Trans 0.1
- Joblet Designs
- Contexts
- Code
- SQL Templates
- Metadata
  - Db Connections
    - localhost 0.1
    - OUTPUT_FINAL 0.1
      - Queries
      - Synonym schemas
      - Table schemas
        - dim_Facility_Type
          - Columns(3)
            - Facility_Key
            - Facility_Type
            - Facility_Name
        - dim_Geo
        - dim_Inspection_Type
        - dim_Results
          - Columns(2)
            - Results_Key
            - Results
        - dim_RiskCategory
        - dim_Violation
      - View schemas
  - File delimited

```
localhost --row1 (Main)-- 740185 rows in 7.9s 93718.03 rows/s --> tMap_1
tMap_1 --ResultsOutput (Main)-- 740185 rows in 7.9s 93718.03 rows/s --> tUniqRow_1
tUniqRow_1 --row2 (Uniques)-- 8 rows in 7.9s 1.01 rows/s --> tMap_2
tMap_2 --Results_Output (Main)-- 8 rows in 8.14s 0.98 rows/s --> OUTPUT_FINAL
```

1 item selected

34°

5:52 PM
2/25/2024



Talend Cloud Data Fabric (R2023-10) | Midterm_Project (Connection: PS)

LOCAL: Midterm_Project

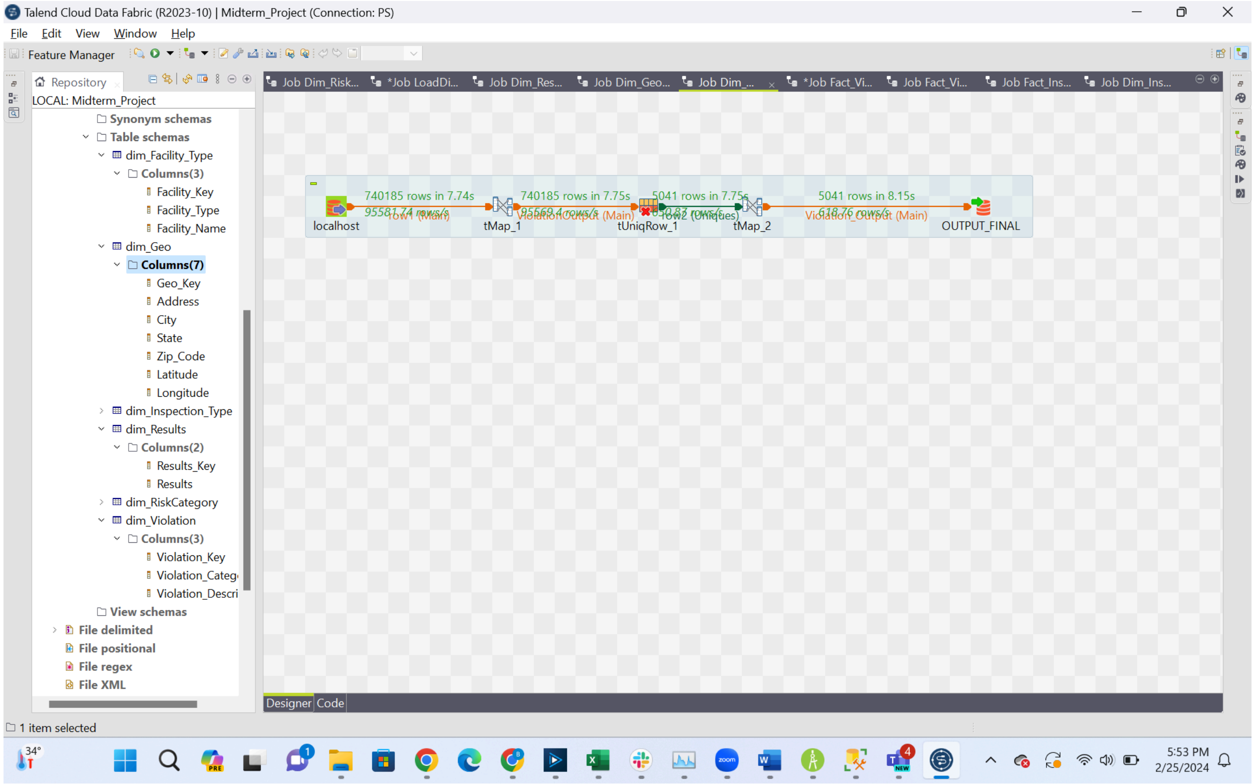
- Dim_Violation 0.1
- Fact_Inspection 0.1
- Fact_Violation 0.1
- Fact_Violations 0.1
- LoadDim_Facility_Type 0.1
- Stage_Table_Trans 0.1
- Joblet Designs
- Contexts
- Code
- SQL Templates
- Metadata
  - Db Connections
    - localhost 0.1
    - OUTPUT_FINAL 0.1
      - Queries
      - Synonym schemas
      - Table schemas
        - dim_Facility_Type
          - Columns(3)
            - Facility_Key
            - Facility_Type
            - Facility_Name
        - dim_Geo
          - Columns(7)
            - Geo_Key
            - Address
            - City
            - State
            - Zip_Code
            - Latitude
            - Longitude
        - dim_Inspection_Type

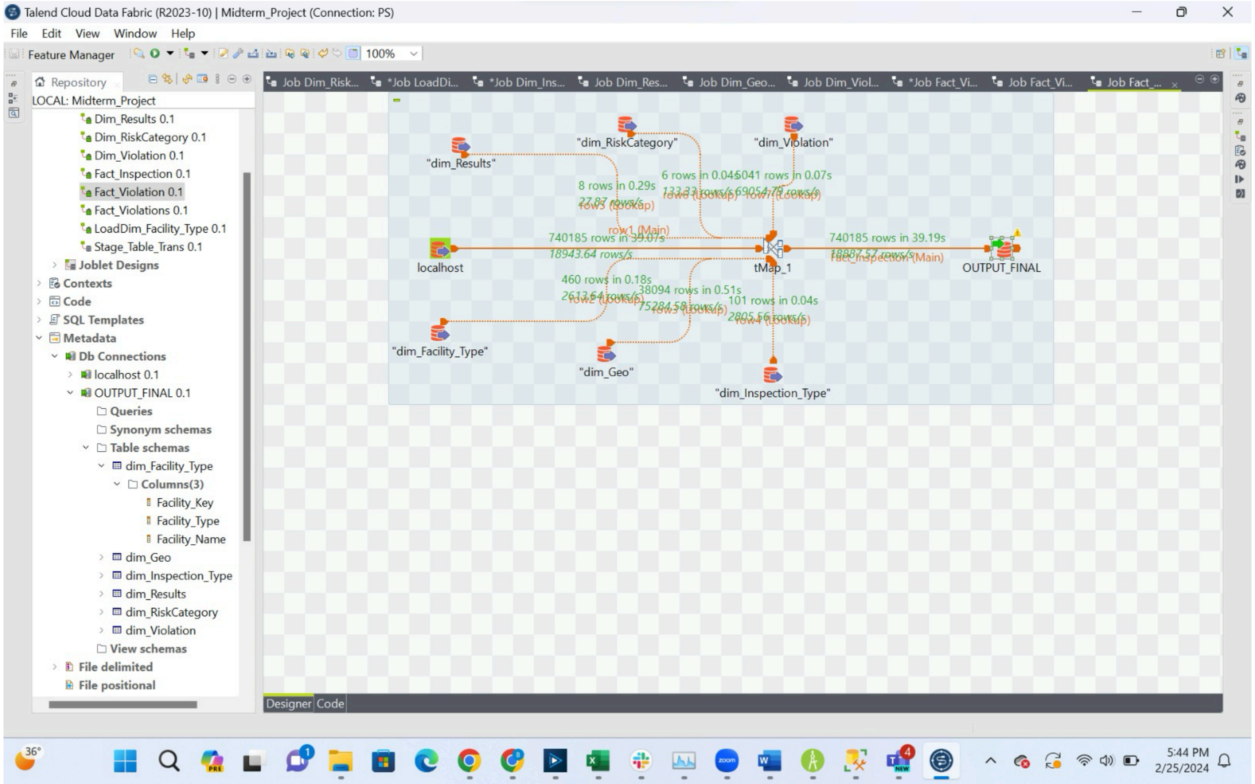
```
localhost --row1 (Main)-- 740185 rows in 10.22s 72425.15 rows/s --> tMap_1
tMap_1 --GeoOut (Main)-- 740185 rows in 10.22s 72425.15 rows/s --> tUniqRow_1
tUniqRow_1 --row2 (Uniques)-- 38094 rows in 10.22s 3727.4 rows/s --> tMap_2
tMap_2 --Geo_Output (Main)-- 38094 rows in 10.68s 3568.52 rows/s --> OUTPUT_FINAL
```

1 item selected

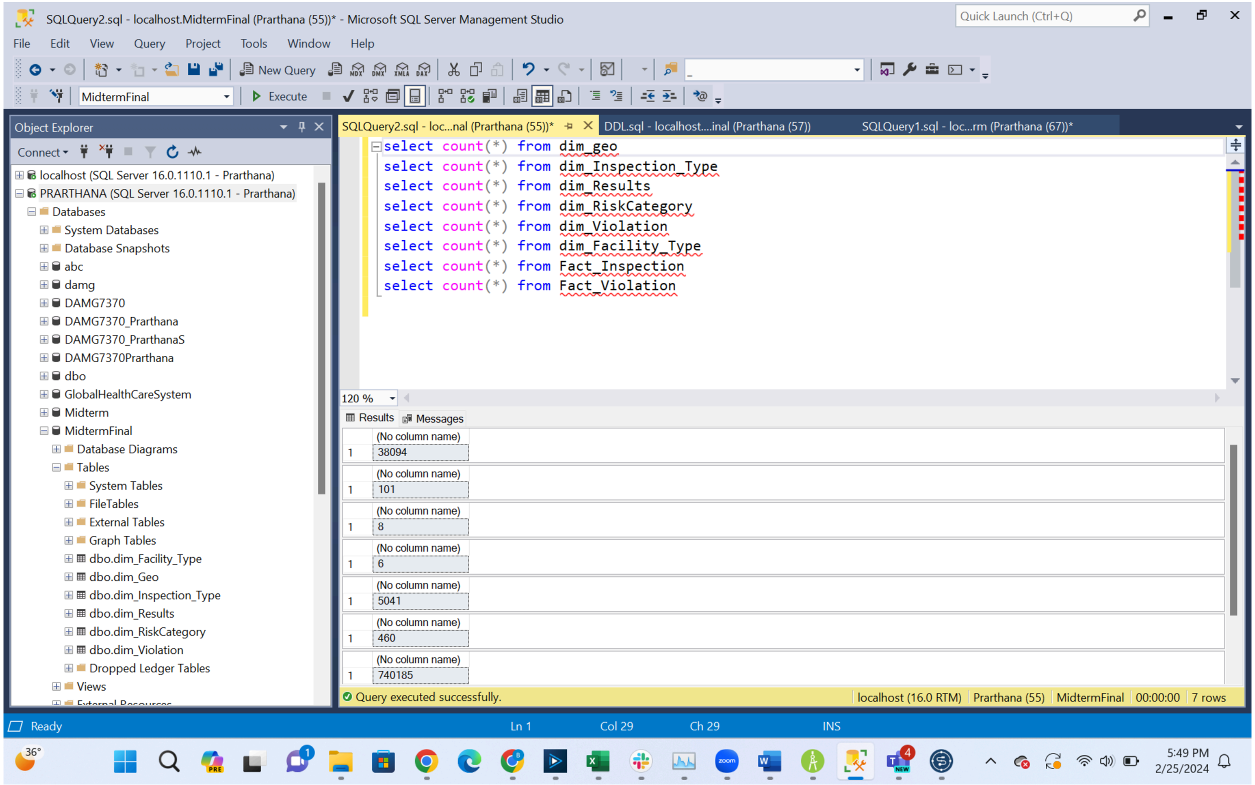
34°

5:53 PM
2/25/2024


Talend Cloud Data Fabric (R2023-10) | Midterm_Project (Connection: PS)
File
Edit
View
Window
Help
Feature Manager
Repository
LOCAL: Midterm_Project
Synonym schemas
Table schemas
dim_Facility_Type
Columns(3)
Facility_Key
Facility_Type
Facility_Name
dim_Geo
Columns(7)
Geo_Key
Address
City
State
Zip_Code
Latitude
Longitude
dim_Inspection_Type
dim_Results
Columns(2)
Results_Key
Results
dim_RiskCategory
dim_Violation
Columns(3)
Violation_Key
Violation_Categ
Violation_Descri
View schemas
File delimited
File positional
File regex
File XML
Job Dim_Risk...
*Job LoadDi...
Job Dim_Res...
Job Dim_Geo...
Job Dim_...
*Job Fact_Vi...
Job Fact_Vi...
Job Fact_Ins...
Job Dim_Ins...
localhost
740185 rows in 7.74s
95581.74 rows/s
row1 (Main)
tMap_1
740185 rows in 7.75s
95569.4 rows/s
ViolationOutput (Main)
tUniqRow_1
5041 rows in 7.75s
650.83 rows/s
row2 (Uniques)
tMap_2
5041 rows in 8.15s
618.76 rows/s
Violation_Output (Main)
OUTPUT_FINAL
Designer
Code
1 item selected
34°
5:53 PM
2/25/2024


Talend Cloud Data Fabric (R2023-10) | Midterm_Project (Connection: PS)
File
Edit
View
Window
Help
Feature Manager
100%
Repository
LOCAL: Midterm_Project
Dim_Results 0.1
Dim_RiskCategory 0.1
Dim_Violation 0.1
Fact_Inspection 0.1
Fact_Violation 0.1
Fact_Violations 0.1
LoadDim_Facility_Type 0.1
Stage_Table_Trans 0.1
Joblet Designs
Contexts
Code
SQL Templates
Metadata
Db Connections
localhost 0.1
OUTPUT_FINAL 0.1
Queries
Synonym schemas
Table schemas
dim_Facility_Type
Columns(3)
Facility_Key
Facility_Type
Facility_Name
dim_Geo
dim_Inspection_Type
dim_Results
dim_RiskCategory
dim_Violation
View schemas
File delimited
File positional
Job Dim_Risk...
*Job LoadDi...
*Job Dim_Ins...
Job Dim_Res...
Job Dim_Geo...
Job Dim_Viol...
*Job Fact_Vi...
Job Fact_Vi...
Job Fact_...
"dim_RiskCategory"
"dim_Violation"
"dim_Results"
8 rows in 0.29s
27.87 rows/s
row5 (Lookup)
6 rows in 0.045
133.33 rows/s
row6 (Lookup)
5041 rows in 0.07s
69054.79 rows/s
row7 (Lookup)
row1 (Main)
740185 rows in 39.07s
18943.64 rows/s
localhost
tMap_1
740185 rows in 39.19s
18887.57 rows/s
Fact_Inspection (Main)
OUTPUT_FINAL
460 rows in 0.18s
2613.64 rows/s
row2 (Lookup)
38094 rows in 0.51s
75284.58 rows/s
row3 (Lookup)
101 rows in 0.04s
2805.56 rows/s
row4 (Lookup)
"dim_Facility_Type"
"dim_Geo"
"dim_Inspection_Type"
Designer
Code
36°
5:44 PM
2/25/2024

Screenshots of dims and facts with row count



```sql
select count(*) from dim_geo
select count(*) from dim_Inspection_Type
select count(*) from dim_Results
select count(*) from dim_RiskCategory
select count(*) from dim_Violation
select count(*) from dim_Facility_Type
select count(*) from Fact_Inspection
select count(*) from Fact_Violation
```

```
   (No column name)
1  38094
   (No column name)
1  101
   (No column name)
1  8
   (No column name)
1  6
   (No column name)
1  5041
   (No column name)
1  460
   (No column name)
1  740185
```

Query executed successfully.

5. **Dashboard**

Business Requirements

a. Examine food inspection results by:

- Inspection result
- Inspection type
- Risk category
- Facility type
- Violations (Codes, descriptions)
- Business inspected
  - DBA (Doing Business As), AKA (Also Known As), License
- Location

b. Inspection report

- All of above with inspection , license , violations & violation description