DAMG 7370 - Designing Advanced Data Architectures for Business Intelligence

MIDTERM PROJECT - Food Inspections of Dallas and Chicago

<u>**Team Members:**</u>

Dharun Karthick Ramaraj

Nitin Sai Varma Indukuri

Nikhil Godalla

Linata Deshmukh

Tools Used:

- **Part-1:**
  - Data Profiling
    - ydata_profiling (Python)
  - Staging:
    - Talend
    - MySQL
- **Part-2:**
  - Dimensional Modeling:
    - E/R Studio Data Architect
- **Part-3:**
  - Loading:
    - Talend
  - Data Warehousing:
    - MySQL
- **Part-4:**
  - Visualizations:
    - PowerBI
    - Tableau

Dataset

Food Inspections

This dataset includes detailed inspections of food establishments in Chicago and Dallas, reflecting efforts to maintain public health and safety standards. In Chicago, inspections are conducted by the Chicago Department of Public Health's Food Protection Program, following a standardized procedure with results reviewed by a State Licensed Environmental Health Practitioner. Data encompasses establishment names, locations, inspection dates, scores, and violations, aiming to enhance transparency and compliance. Similarly, the Dallas dataset is provided by the Code Compliance Services Department's Consumer Health Division, ensuring thorough evaluations and adherence to safety protocols. Both datasets represent the cities' commitment to upholding food safety, offering critical insights into the operational standards of food service providers in Chicago and Dallas.

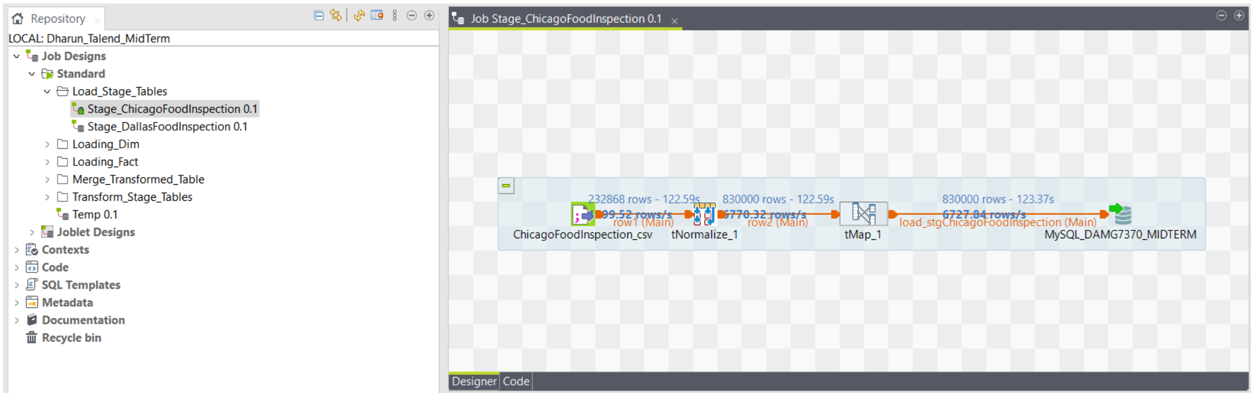
Restaurant and Food Establishment Inspections (October 2016 to Present)

This data set is intended to communicate the name of establishment, the physical location of the establishment, the date the inspection was conducted, the overall score for the inspection, and the point deduction for the individual violations.

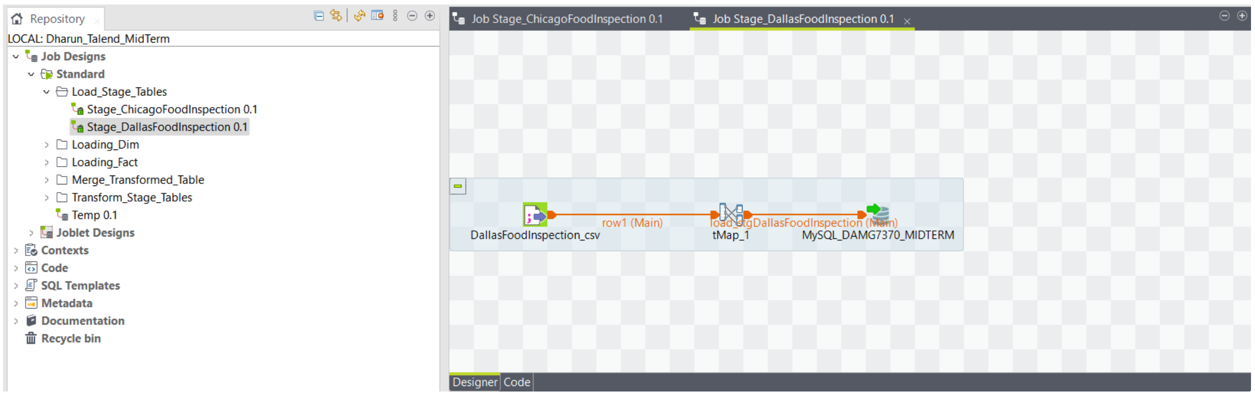
Deliverables

Screenshot of making Stage Tables

Staging Chicago



Staging Dallas



List total time your job took to complete (Runtimes).

DDL script for Stage table(s).

Chicago:

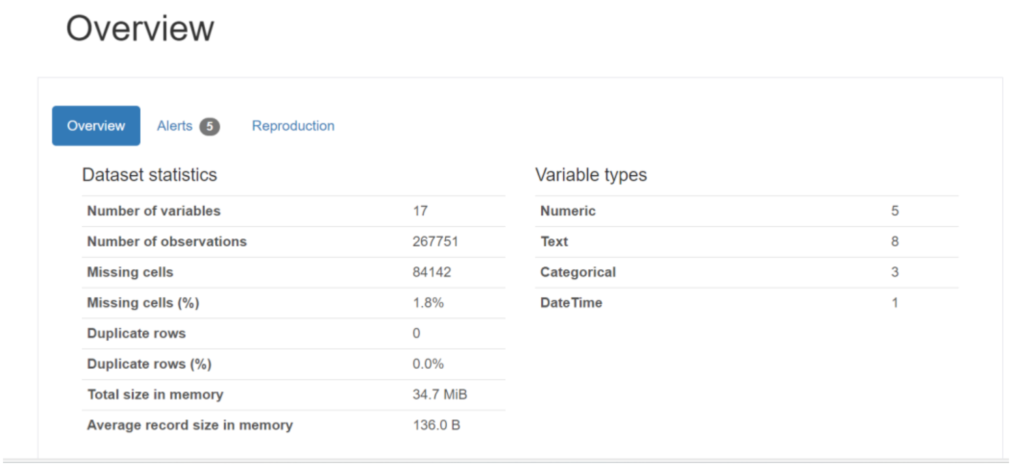
```
CREATE TABLE `stg_chicago_foodinspection` (
  `ChicagoUID` int NOT NULL,
  `Inspection_ID` int DEFAULT NULL,
  `DBA_Name` varchar(100) DEFAULT NULL,
  `AKA_Name` varchar(100) DEFAULT NULL,
  `License` int DEFAULT NULL,
  `Facility_Type` varchar(200) DEFAULT NULL,
  `Risk` varchar(20) DEFAULT NULL,
  `Address` varchar(200) DEFAULT NULL,
  `City` varchar(200) DEFAULT NULL,
  `State` varchar(200) DEFAULT NULL,
  `Zip` varchar(200) DEFAULT NULL,
  `Inspection_Date` datetime DEFAULT NULL,
  `Inspection_Type` varchar(50) DEFAULT NULL,
  `Results` varchar(50) DEFAULT NULL,
  `Violations` varchar(6000) DEFAULT NULL,
  `Latitude` varchar(20) DEFAULT NULL,
  `Longitude` varchar(20) DEFAULT NULL,
  `Location` varchar(50) DEFAULT NULL,
  `DI_CreateDate` datetime DEFAULT NULL,
  `DI_WorkflowFileName` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`ChicagoUID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Dallas:

```
CREATE TABLE `stg_dallas_foodinspection` (
  `DallasUID` int NOT NULL,
  `Restaurant_Name` varchar(65) DEFAULT NULL,
  `Inspection_Type` varchar(9) DEFAULT NULL,
  `Inspection_Date` datetime DEFAULT NULL,
  `Inspection_Score` int DEFAULT NULL,
  `Street_Number` int DEFAULT NULL,
  `Street_Name` varchar(25) DEFAULT NULL,
  `Street_Direction` varchar(3) DEFAULT NULL,
  `Street_Type` varchar(4) DEFAULT NULL,
  `Street_Unit` varchar(5) DEFAULT NULL,
  `Street_Address` varchar(37) DEFAULT NULL,
  `Zip_Code` varchar(10) DEFAULT NULL,
  `Violation_Description___1` varchar(101) DEFAULT NULL,
  `Violation_Points___1` varchar(3) DEFAULT NULL,
  `Violation_Description___2` varchar(100) DEFAULT NULL,
  `Violation_Points___2` varchar(3) DEFAULT NULL,
  `Violation_Description___3` varchar(100) DEFAULT NULL,
  `Violation_Points___3` varchar(3) DEFAULT NULL,
  `Violation_Description___4` varchar(100) DEFAULT NULL,
  `Violation_Points___4` varchar(3) DEFAULT NULL,
  `Violation_Description___5` varchar(100) DEFAULT NULL,
  `Violation_Points___5` varchar(3) DEFAULT NULL,
  `Violation_Description___6` varchar(100) DEFAULT NULL,
  `Violation_Points___6` varchar(3) DEFAULT NULL,
  `Violation_Description___7` varchar(100) DEFAULT NULL,
  `Violation_Points___7` varchar(3) DEFAULT NULL,
  `Violation_Description___8` varchar(100) DEFAULT NULL,
  `Violation_Points___8` varchar(3) DEFAULT NULL,
  `Violation_Description___9` varchar(100) DEFAULT NULL,
  `Violation_Points___9` varchar(3) DEFAULT NULL,
  `Violation_Description___10` varchar(100) DEFAULT NULL,
  `Violation_Points___10` varchar(3) DEFAULT NULL,
  `Violation_Description___11` varchar(100) DEFAULT NULL,
  `Violation_Points___11` varchar(3) DEFAULT NULL,
  `Violation_Description___12` varchar(100) DEFAULT NULL,
  `Violation_Points___12` varchar(3) DEFAULT NULL,
  `Violation_Description___13` varchar(100) DEFAULT NULL,
  `Violation_Points___13` varchar(3) DEFAULT NULL,
  `Violation_Description___14` varchar(100) DEFAULT NULL,
  `Violation_Points___14` varchar(3) DEFAULT NULL,
  `Violation_Description___15` varchar(100) DEFAULT NULL,
  `Violation_Points___15` varchar(3) DEFAULT NULL,
  `Violation_Description___16` varchar(100) DEFAULT NULL,
  `Violation_Points___16` varchar(3) DEFAULT NULL,
  `Violation_Description___17` varchar(100) DEFAULT NULL,
  `Violation_Points___17` varchar(3) DEFAULT NULL,
  `Violation_Description___18` varchar(100) DEFAULT NULL,
  `Violation_Points___18` varchar(3) DEFAULT NULL,
  `Violation_Description___19` varchar(100) DEFAULT NULL,
  `Violation_Points___19` varchar(3) DEFAULT NULL,
  `Violation_Description___20` varchar(100) DEFAULT NULL,
  `Violation_Points___20` varchar(3) DEFAULT NULL,
  `Violation_Description___21` varchar(100) DEFAULT NULL,
  `Violation_Points___21` varchar(3) DEFAULT NULL,
  `Violation_Description___22` varchar(100) DEFAULT NULL,
  `Violation_Points___22` varchar(3) DEFAULT NULL,
  `Violation_Description___23` varchar(100) DEFAULT NULL,
  `Violation_Points___23` varchar(3) DEFAULT NULL,
  `Violation_Description___24` varchar(59) DEFAULT NULL,
  `Violation_Points___24` varchar(3) DEFAULT NULL,
  `Violation_Description___25` varchar(50) DEFAULT NULL,
  `Violation_Points___25` varchar(3) DEFAULT NULL,
  `Inspection_Month` varchar(8) DEFAULT NULL,
  `Inspection_Year` varchar(6) DEFAULT NULL,
  `Latitute` varchar(150) DEFAULT NULL,
  `Longitute` varchar(150) DEFAULT NULL,
  `DI_CreateDate` datetime DEFAULT NULL,
  `DI_WorkflowFileName` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`DallasUID`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Analysis report on Data profiling

State : Chicago



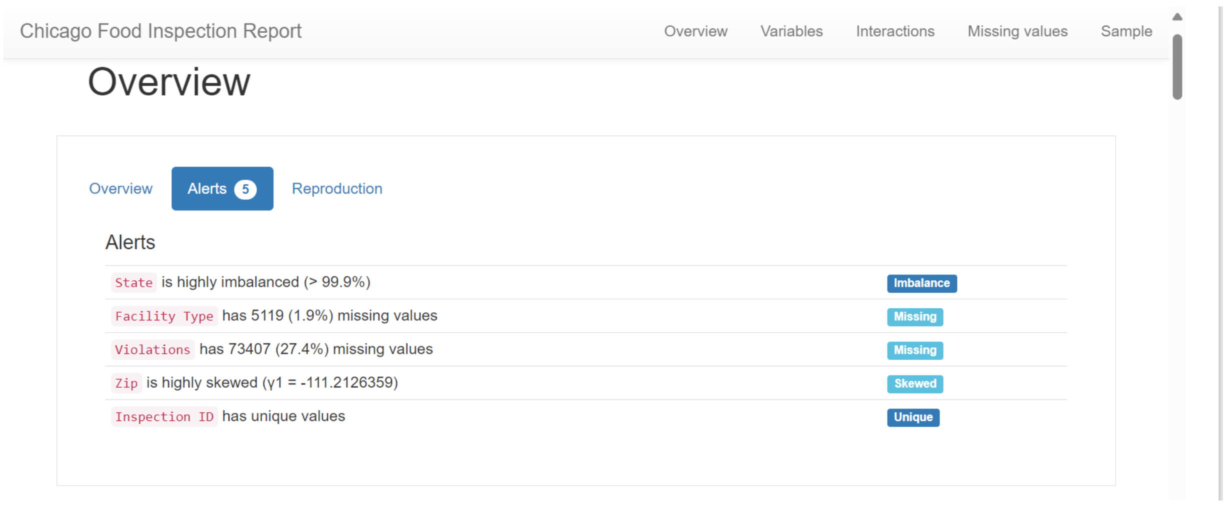
Overview

Dataset statistics

Number of variables	17
Number of observations	267751
Missing cells	84142
Missing cells (%)	1.8%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	34.7 MiB
Average record size in memory	136.0 B

Variable types

Numeric	5
Text	8
Categorical	3
DateTime	1



Chicago Food Inspection Report

Overview

Alerts

`State` is highly imbalanced (> 99.9%)	Imbalance
`Facility Type` has 5119 (1.9%) missing values	Missing
`Violations` has 73407 (27.4%) missing values	Missing
`Zip` is highly skewed (γ1 = -111.2126359)	Skewed
`Inspection ID` has unique values	Unique

The data profiling analysis of the Chicago Food Inspection Report reveals the following key points to be addressed:

Multi-Valued Attributes: The 'Violations' column, which contains 27.4% missing values, likely holds multiple pieces of information per entry. These will be split into separate, atomic data fields such as 'Violation Code', 'Violation Description', and 'Violation Comments' to align with the principles of database normalization.

Data Type Conversions:

Numeric fields will be assessed for appropriate data types (integer or float) based on the range and precision required.

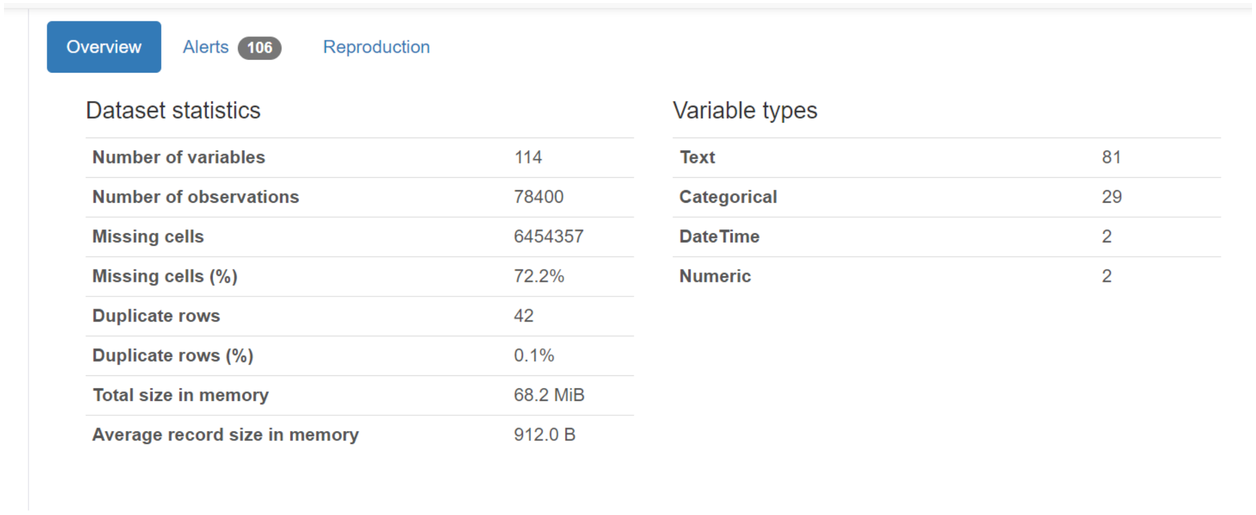
Text fields will be checked for consistency and converted to categorical where appropriate, to optimize storage and querying.

The highly skewed 'Zip' field will be evaluated to determine if it should be treated as a categorical variable rather than numeric, which can improve analysis given its skewness.

Data Imbalance: The imbalance in the 'State' field will be addressed by analyzing the distribution and considering data collection strategies to balance the dataset if used for predictive modeling.

Missing Values: Strategies for handling missing values in 'Facility Type' and 'Violations' will include data imputation techniques or excluding these records from certain analyses, depending on the context and necessity of complete information for the analysis.

State : Dallas

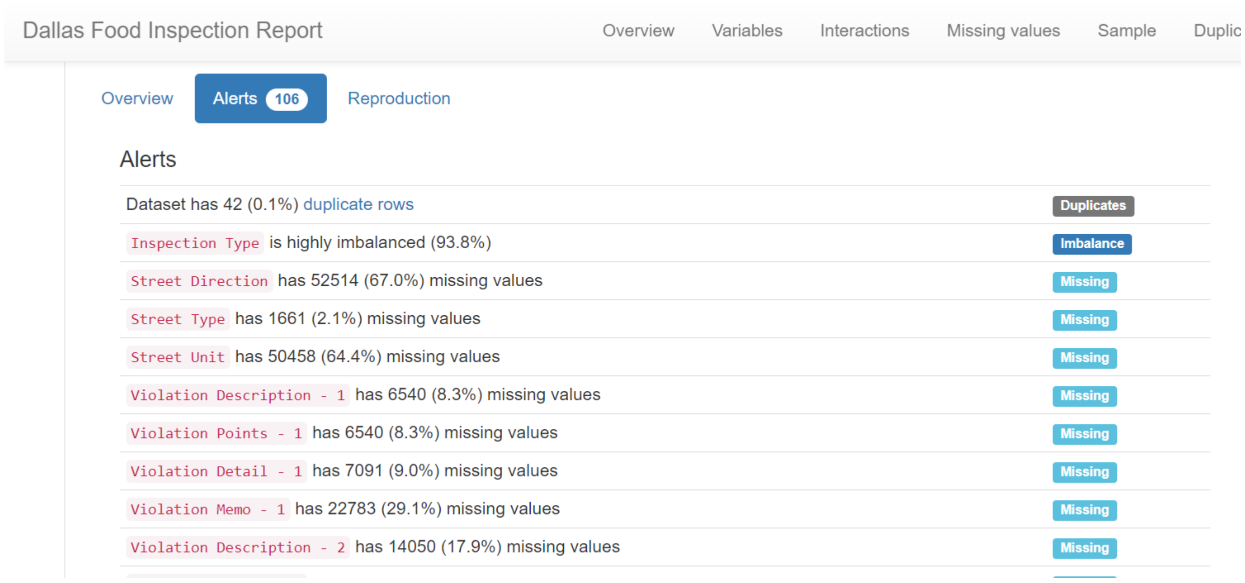


Dataset statistics

Number of variables	114
Number of observations	78400
Missing cells	6454357
Missing cells (%)	72.2%
Duplicate rows	42
Duplicate rows (%)	0.1%
Total size in memory	68.2 MiB
Average record size in memory	912.0 B

Variable types

Text	81
Categorical	29
DateTime	2
Numeric	2



Alerts

Dataset has 42 (0.1%) duplicate rows	Duplicates
`Inspection Type` is highly imbalanced (93.8%)	Imbalance
`Street Direction` has 52514 (67.0%) missing values	Missing
`Street Type` has 1661 (2.1%) missing values	Missing
`Street Unit` has 50458 (64.4%) missing values	Missing
`Violation Description - 1` has 6540 (8.3%) missing values	Missing
`Violation Points - 1` has 6540 (8.3%) missing values	Missing
`Violation Detail - 1` has 7091 (9.0%) missing values	Missing
`Violation Memo - 1` has 22783 (29.1%) missing values	Missing
`Violation Description - 2` has 14050 (17.9%) missing values	Missing

The data profiling analysis of the Dallas Food Inspection Report reveals the following key points to be addressed:

Multi-Valued Attributes:

Identify and separate distinct values within multi-valued attributes into individual columns or create a relational table that can be joined with the main dataset on a unique identifier.

For variables like 'Violation Descriptions,' where multiple violations may be listed within a single record, we will unpack these into separate columns or rows, depending on the analysis requirement, ensuring that each violation is distinctly represented.

Data Type Conversions:

Text and categorical variables will be reviewed to determine if conversion to a numeric format is necessary, for instance, through encoding techniques such as one-hot encoding for categorical variables.

DateTime variables will be checked for consistency and converted into a standard datetime format to facilitate time-series analysis or chronological sorting.

Numeric variables will be examined for scale and distribution, and normalized or standardized if required for analytical algorithms that are sensitive to variable scales.

Handling Missing Values:

For variables with a high percentage of missing data, such as 'Street Direction' (67% missing) and 'Street Unit' (64.4% missing), we will consider imputation, where appropriate, or exclusion if the variable is not critical to the analysis.

In the case of 'Violation' fields, given the high variability in missing data (ranging from 8.3% to over 99.9%), a threshold for missing percentage may be set, above which the variable may be excluded from analysis.

We will also explore the pattern of missingness to determine if it is random or if there's an underlying pattern that could inform the imputation strategy.

Dealing with Data Imbalance:

The imbalance in the 'Inspection Type' variable, where 93.8% of the data is skewed towards specific categories, will be addressed through resampling techniques. Oversampling of the minority class or undersampling of the majority class could be considered, depending on the analysis objectives.

Alternatively, if resampling is not appropriate, advanced algorithms that are robust to class imbalance, such as tree-based methods, may be utilized.

Dealing with Duplicates:

The 42 duplicate rows identified will be removed to ensure that the dataset does not contain redundant information, which could bias the analysis.

Addressing Multi-Valued Attributes

Food_Inspections_20240222 - Dallas

In the Dallas inspection report, each violation entry is multivalued, consisting of a description, points, detail, and memo. These entries are indexed numerically for each inspection. This structure presents a challenge for relational databases, which are better suited for single-valued attributes, and would typically require normalization to separate these multivalued attributes into distinct tables linked by foreign keys for effective data management and analysis.

Restaurant and Food Establishment Inspections (October 2016 to Present)- Chicago

The data in each cell of a health inspection report is organized to convey specific information about compliance with regulations. It includes:

Violation Number: A code identifying the regulation violated.
Violation Description: A summary of the regulatory requirement.
Comments: Detailed observations and instructions for correction.

The columns for violation details are multi-valued are addressed in the following ways:

- Parsing**:** Text parsing techniques to separate the different components within the cell. Regular expressions or delimiter-based splitting are used if there are consistent patterns or symbols separating the values.
- Normalization: A DIM_Violation table has been created which holds unique violation types. This table includes:

  ViolationSK: A surrogate key that uniquely identifies each violation type.
  DI_CreateDate: The date the record was created in the dimension table.
  DI_WorkflowFileName: The name of the workflow file associated with the creation of this record.
  ViolationDescription: A textual description of the violation.
  ViolationCode: A unique code for each type of violation.

  This table ensures that each type of violation is only stored once, regardless of how many times it occurs across different inspections. The ViolationSK serves as a unique identifier for each type of violation.

- Bridge Table (FACT_ViolationInspection): The FACT_ViolationInspection table acts as a bridge between inspections and violations and handles the many-to-many relationship. It includes:

  - BridgeSK: A surrogate key that uniquely identifies each record in the bridge table.
  - InspectionSK: A foreign key linking to the FACT_FoodInspection table, identifying the inspection.
  - ViolationSK: A foreign key linking to the DIM_Violation table, identifying the violation that occurred.
  - ViolationPoints: The points associated with the violation for this inspection.
  - ViolationComments: Any comments or memos associated with the violation for this inspection.

  This bridge table allows for recording each occurrence of a violation in relation to an inspection. A single inspection can have multiple violations, and a single violation type can be associated with multiple inspections.

- FACT_FoodInspection: The FACT_FoodInspection table holds the inspection records.

  It includes:

  - InspectionSK: A surrogate key that uniquely identifies each inspection.
  - Inspection Result: The outcome of the inspection.
  - RiskSK: A foreign key linking to a table that categorizes the level of risk.
  - TotalViolationScore: A cumulative score of the violations associated with the inspection.
  - DI_CreateDate, InspectionID, DI_WorkflowFileName: Metadata about the inspection record.

- Inspection TypeSK, FoodPlacesSK, DateSK: Foreign keys linking to other dimension tables. This design allows for a flexible and scalable solution to handle multi-valued attributes by separating the static data about violations into its own dimension table (DIM_Violation) and recording each instance of a violation occurring in an inspection in the bridge table (FACT_ViolationInspection). Each inspection's unique details are kept in the FACT_FoodInspection table. This structure facilitates complex querying and analysis, as it enables you to track the frequency of specific violations, the total points for violations per inspection, and other aggregate statistics without redundancy and while maintaining referential integrity.

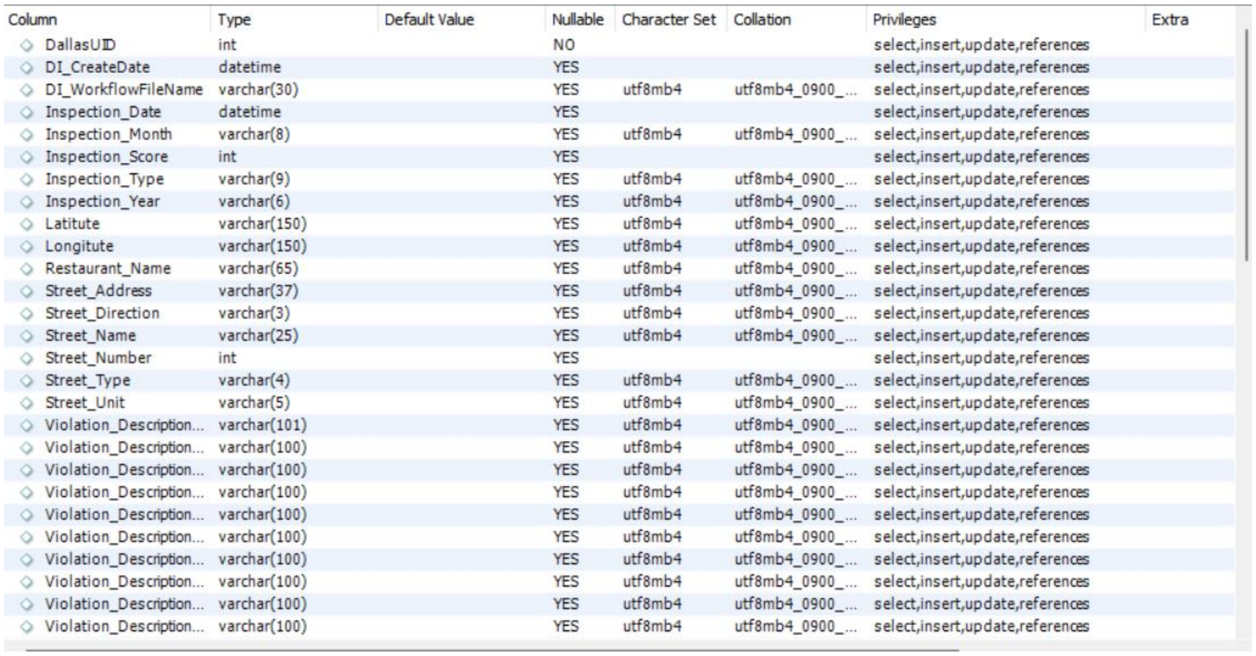
Datatype Conversions

- tMap component along with the TalendDate.getCurrentDate() function to transform string date representations into Date objects, ensuring proper date pattern matching and error handling for a seamless data type conversion.

Schema differences between the dataset and plan to merge the data.

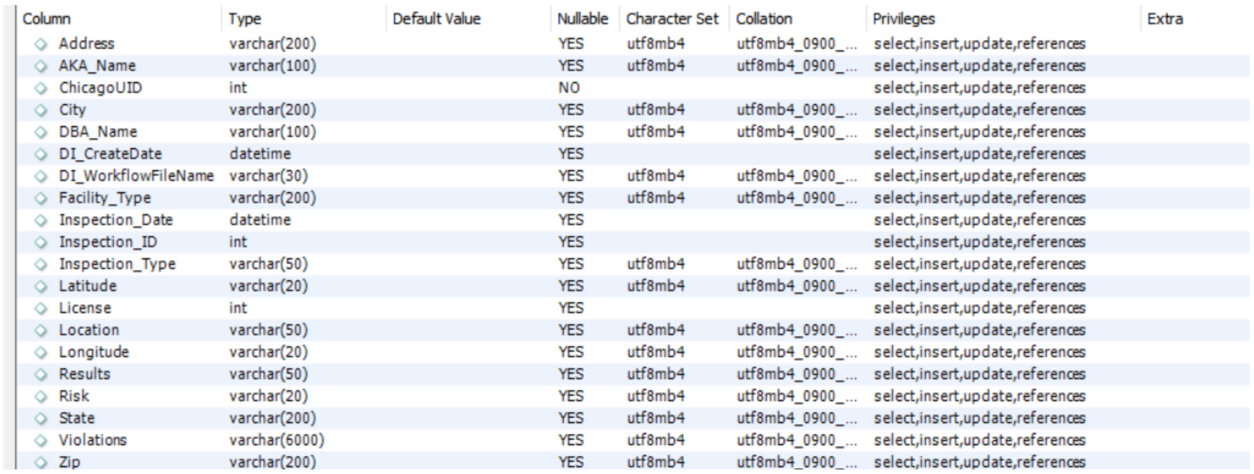
After Staging:

Dallas Dataset Staging Schema



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
DallasUID	int		NO			select,insert,update,references	
DI_CreateDate	datetime		YES			select,insert,update,references	
DI_WorkflowFileName	varchar(30)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Inspection_Date	datetime		YES			select,insert,update,references	
Inspection_Month	varchar(8)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Inspection_Score	int		YES			select,insert,update,references	
Inspection_Type	varchar(9)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Inspection_Year	varchar(6)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Latitute	varchar(150)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Longitute	varchar(150)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Restaurant_Name	varchar(65)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Street_Address	varchar(37)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Street_Direction	varchar(3)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Street_Name	varchar(25)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Street_Number	int		YES			select,insert,update,references	
Street_Type	varchar(4)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Street_Unit	varchar(5)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(101)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Description...	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	

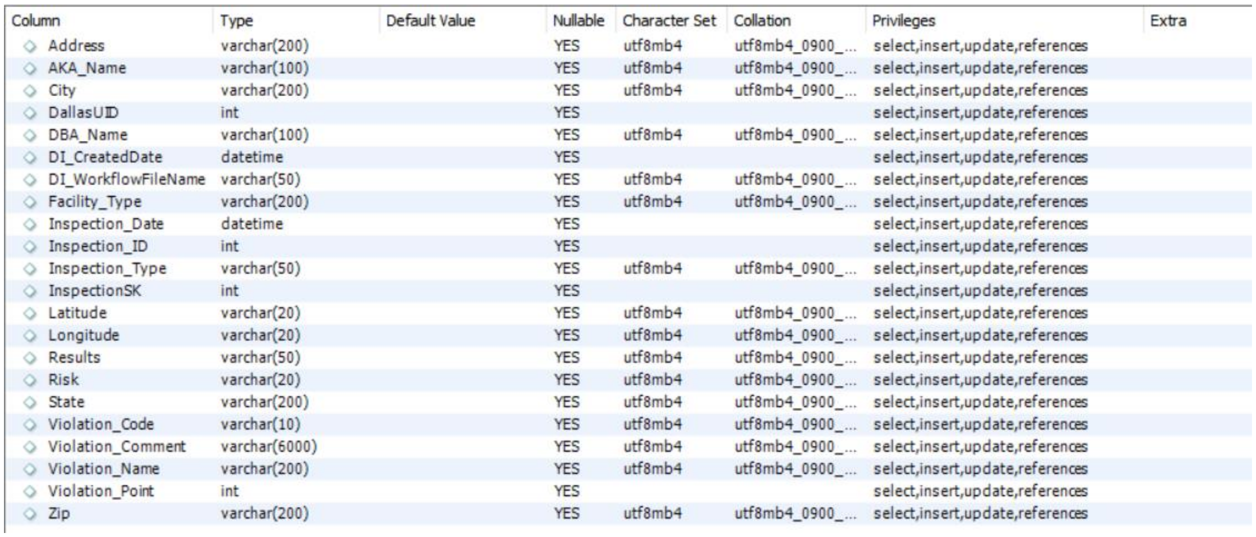
Chicago Dataset Staging Schema



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
Address	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
AKA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
ChicagoUID	int		NO			select,insert,update,references	
City	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DBA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DI_CreateDate	datetime		YES			select,insert,update,references	
DI_WorkflowFileName	varchar(30)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Facility_Type	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Inspection_Date	datetime		YES			select,insert,update,references	
Inspection_ID	int		YES			select,insert,update,references	
Inspection_Type	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Latitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
License	int		YES			select,insert,update,references	
Location	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Longitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Results	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Risk	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
State	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violations	varchar(6000)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Zip	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	

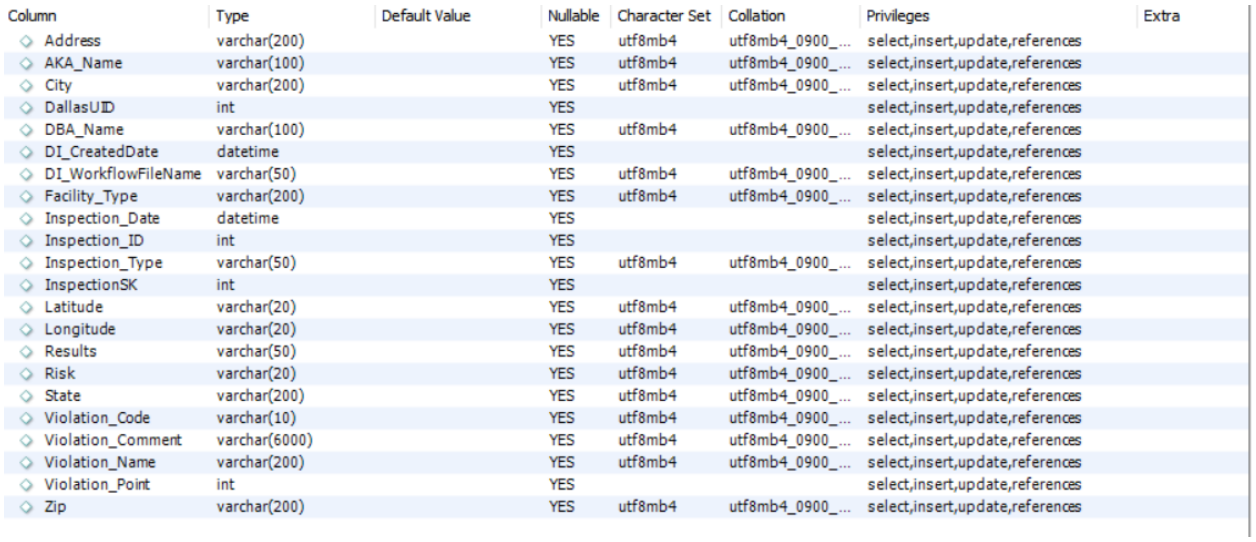
After Loading:

Dallas Dataset Loading Schema



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
Address	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
AKA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
City	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DallasUID	int		YES			select,insert,update,references	
DBA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DI_CreatedDate	datetime		YES			select,insert,update,references	
DI_WorkflowFileName	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Facility_Type	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Inspection_Date	datetime		YES			select,insert,update,references	
Inspection_ID	int		YES			select,insert,update,references	
Inspection_Type	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
InspectionSK	int		YES			select,insert,update,references	
Latitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Longitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Results	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Risk	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
State	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Code	varchar(10)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Comment	varchar(6000)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Name	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Point	int		YES			select,insert,update,references	
Zip	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	

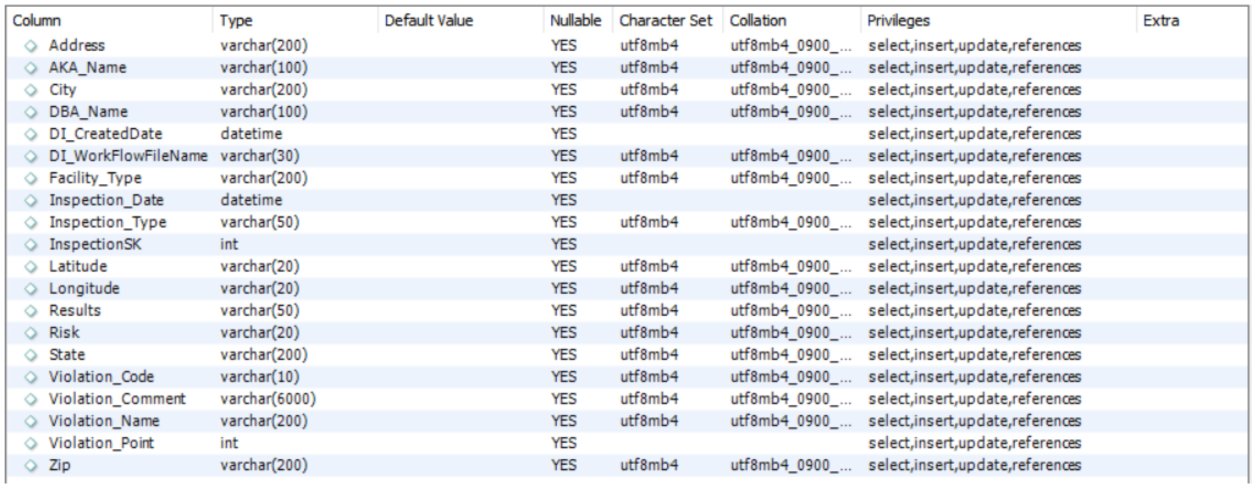
Chicago Dataset Loading Schema



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
Address	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
AKA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
City	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DallasUID	int		YES			select,insert,update,references	
DBA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DI_CreatedDate	datetime		YES			select,insert,update,references	
DI_WorkflowFileName	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Facility_Type	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Inspection_Date	datetime		YES			select,insert,update,references	
Inspection_ID	int		YES			select,insert,update,references	
Inspection_Type	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
InspectionSK	int		YES			select,insert,update,references	
Latitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Longitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Results	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Risk	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
State	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Code	varchar(10)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Comment	varchar(6000)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Name	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Point	int		YES			select,insert,update,references	
Zip	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	

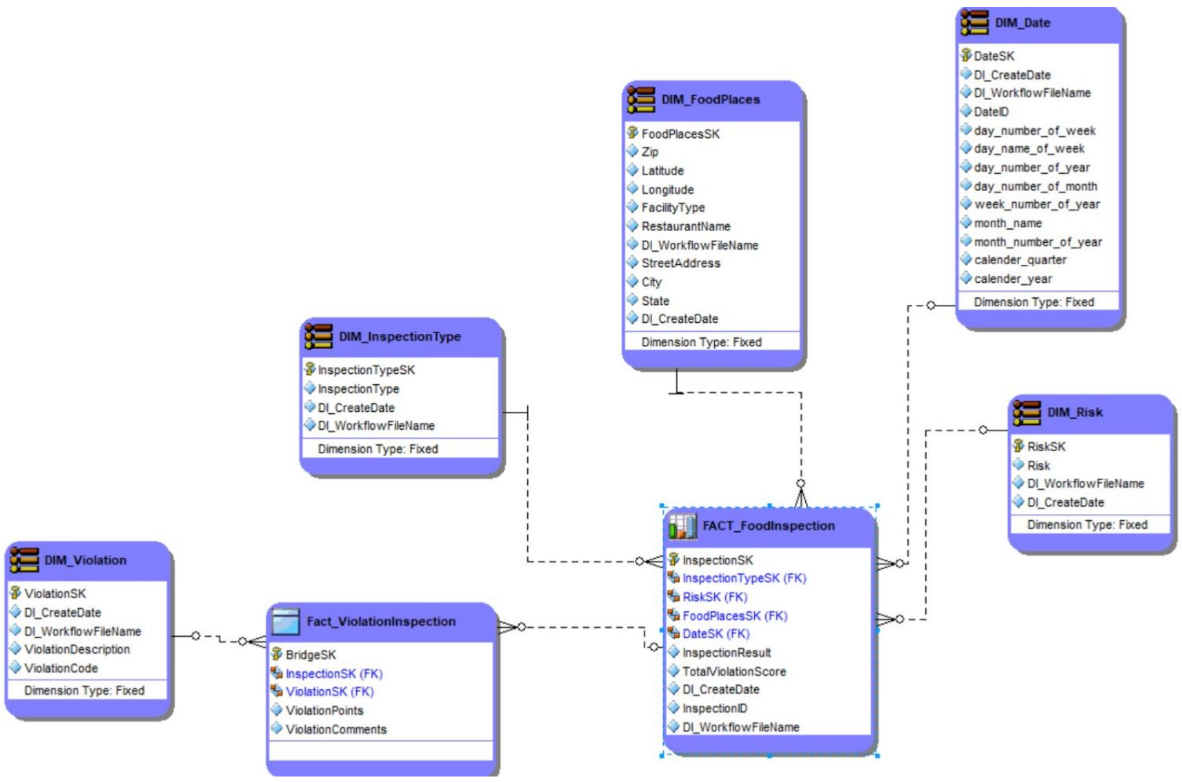
Merging:

Merged Dallas and Chicago Dataset Schema



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
Address	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
AKA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
City	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DBA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DI_CreatedDate	datetime		YES			select,insert,update,references	
DI_WorkFlowFileName	varchar(30)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Facility_Type	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Inspection_Date	datetime		YES			select,insert,update,references	
Inspection_Type	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
InspectionSK	int		YES			select,insert,update,references	
Latitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Longitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Results	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Risk	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
State	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Code	varchar(10)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Comment	varchar(6000)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Name	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Point	int		YES			select,insert,update,references	
Zip	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	

Description of Dimensional Data Model (ER/Studio).



Fact Tables:

- **FACT_FoodInspection**: This is a central fact table that contains quantitative data (measures) for each food inspection.

  Attributes: InspectionSK, InspectionResult, RiskSK (foreign key), TotalViolationScore, DI_CreateDate, InspectionID, DI_WorkflowFileName, InspectionTypeSK (foreign key), FoodPlacesSK (foreign key), DateSK (foreign key).

- **FACT_ViolationInspection**: This is a bridge table that connects inspections to violations, indicating a many-to-many relationship.

  Attributes: BridgeSK, InspectionSK (foreign key), ViolationSK (foreign key), ViolationPoints, ViolationComments.

Dimension Tables:

- **DIM_Date**: This dimension table provides a date hierarchy for time-based analysis.

  Attributes: DateSK, DI_CreateDate, DI_WorkflowFileName, full_date_ak, day_number_of_week, day_name_of_week, day_number_of_year, day_number_of_month, week_number_of_year, month_name, month_number_of_year, calendar_quarter, calendar_year.

- **DIM_Risk**: This dimension table categorizes inspections by risk level.

  Attributes: RiskSK, Risk, DI_WorkflowFileName, DI_CreateDate.

- **DIM_InspectionType**: This dimension table categorizes the type of inspections being performed.

Attributes: InspectionTypeSK, InspectionType, DI_CreateDate, DI_WorkflowFileName.

- **DIM_Violation**: This dimension table details the types of violations that can be found during an inspection.

  Attributes: ViolationSK, DI_CreateDate, DI_WorkflowFileName, ViolationDescription, ViolationCode.

- **DIM_FoodPlaces**: Although not listed in your text, it is visible in the image and likely contains information about the places where food inspections are conducted.

  Attributes: FoodPlacesSK, Zip, Latitude, Longitude, FacilityType, RestaurantName, DI_WorkflowFileName, StreetAddress, City, State, DI_CreateDate.

Dimensional Data Model:

- **Dimensions**: These are descriptive attributes related to the fact data and are used to filter and group data in the fact tables for analysis. They are usually textual fields and describe various aspects like time (DIM_Date), risk level (DIM_Risk), types of inspections (DIM_InspectionType), and the specifics of violations (DIM_Violation).
- **Facts**: These tables contain the performance metrics or KPIs that businesses are interested in analyzing. The fact tables typically have foreign keys that correspond to primary keys in the dimension tables, creating a star schema or snowflake schema.

This model allows for complex analytical queries, enabling users to perform trend analysis over time, by inspection type, by risk level, and to dive into the details of specific violations. The clear separation of dimensions and facts also enables efficient storage and high-speed retrieval, which are key performance considerations in data warehousing and business intelligence.

Dimesnional Model DDL Scripts

Bridge_Violations

```
CREATE TABLE `bridge_violation` (
  `BridgeSK` int NOT NULL,
  `InspectionSK` int DEFAULT NULL,
  `Violation_SK` int NOT NULL,
  `Violation_Point` int DEFAULT NULL,
  `Violation_Comment` varchar(6000) DEFAULT NULL,
  `DI_CreatedDate` datetime DEFAULT NULL,
  `DI_WorkFlowFileName` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`BridgeSK`,`Violation_SK`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Dim_Date

```
CREATE TABLE `dim_date` (
  `DateSK` int NOT NULL,
  `DateID` date DEFAULT NULL,
  `day_number_of_week` int DEFAULT NULL,
  `day_name_of_week` varchar(30) DEFAULT NULL,
  `day_number_of_year` int DEFAULT NULL,
  `day_number_of_month` int DEFAULT NULL,
  `week_number_of_year` int DEFAULT NULL,
  `month_name` varchar(30) DEFAULT NULL,
  `month_number_of_year` int DEFAULT NULL,
  `calendar_quarter` int DEFAULT NULL,
  `calendar_year` varchar(30) DEFAULT NULL,
  `DI_CreateDate` datetime DEFAULT NULL,
  `DI_WorkflowFileName` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`DateSK`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Dim_foodplaces

```
CREATE TABLE `dim_foodplaces` (
  `FoodPlacesSK` int NOT NULL,
  `RestaurantName` varchar(100) DEFAULT NULL,
  `AK_A_Name` varchar(100) DEFAULT NULL,
  `Facility_Type` varchar(200) DEFAULT NULL,
  `StreetAddress` varchar(200) DEFAULT NULL,
  `Zip` varchar(200) DEFAULT NULL,
  `City` varchar(200) DEFAULT NULL,
  `State` varchar(200) DEFAULT NULL,
  `Latitude` varchar(30) DEFAULT NULL,
  `Longitude` varchar(30) DEFAULT NULL,
  `DI_CreateDate` datetime DEFAULT NULL,
  `DI_WorkflowFileName` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`FoodPlacesSK`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Dim_inspectiontype

```
CREATE TABLE `dim_inspectiontype` (
  `InspectionTypeSK` int NOT NULL,
  `Inspection_Type` varchar(50) DEFAULT NULL,
  `DI_CreatedDate` datetime DEFAULT NULL,
  `DI_WorkFlowFileName` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`InspectionTypeSK`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Dim_risk

```
CREATE TABLE `dim_risk` (
  `RiskSK` int NOT NULL,
  `Risk` varchar(20) DEFAULT NULL,
  `DI_CreatedDate` datetime DEFAULT NULL,
  `DI_WorkFlowFileName` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`RiskSK`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Dim_violations

```
CREATE TABLE `dim_violation` (
  `Violation_SK` int NOT NULL,
  `Violation_Code` varchar(10) DEFAULT NULL,
  `Violation_Name` varchar(200) DEFAULT NULL,
  `DI_CreatedDate` datetime DEFAULT NULL,
  `DI_WorkFlowFileName` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`Violation_SK`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

fact_inspection

```
CREATE TABLE `fact_inspection` (
  `InspectionSK` int DEFAULT NULL,
  `DateSK` int DEFAULT NULL,
  `FoodPlacesSK` int DEFAULT NULL,
  `RiskSK` int DEFAULT NULL,
  `InspectionTypeSK` int DEFAULT NULL,
  `TotalViolationScore` int DEFAULT NULL,
  `InspectionResult` varchar(30) DEFAULT NULL,
  `DI_CreatedDate` datetime DEFAULT NULL,
  `DI_WorkflowFileName` varchar(40) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci
```

Data into Integration Schema

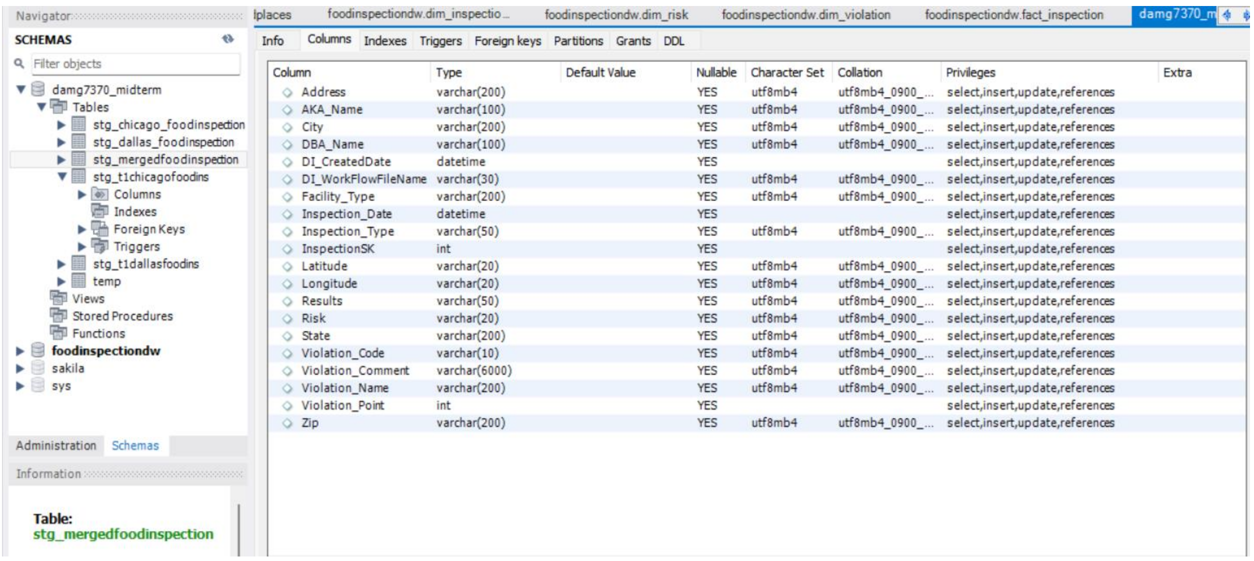
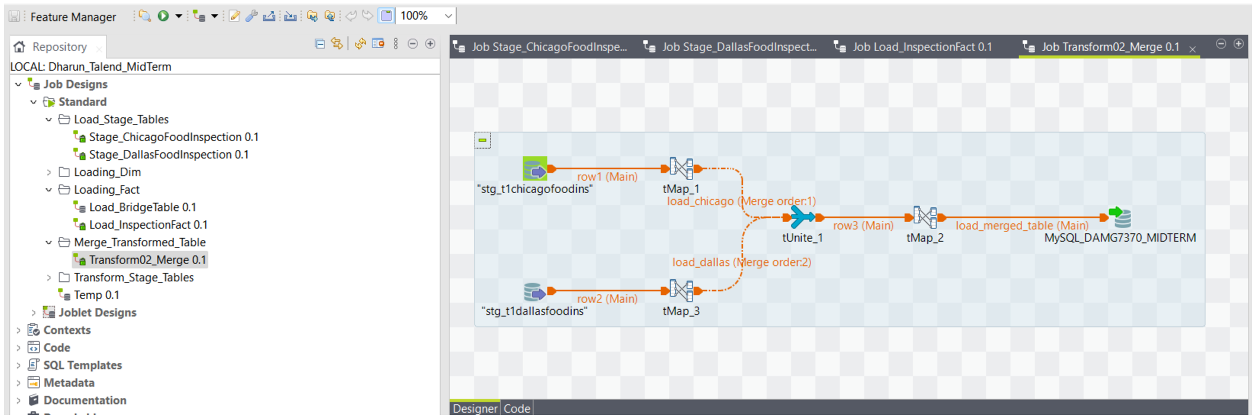


Table:
stg_mergedfoodinspection

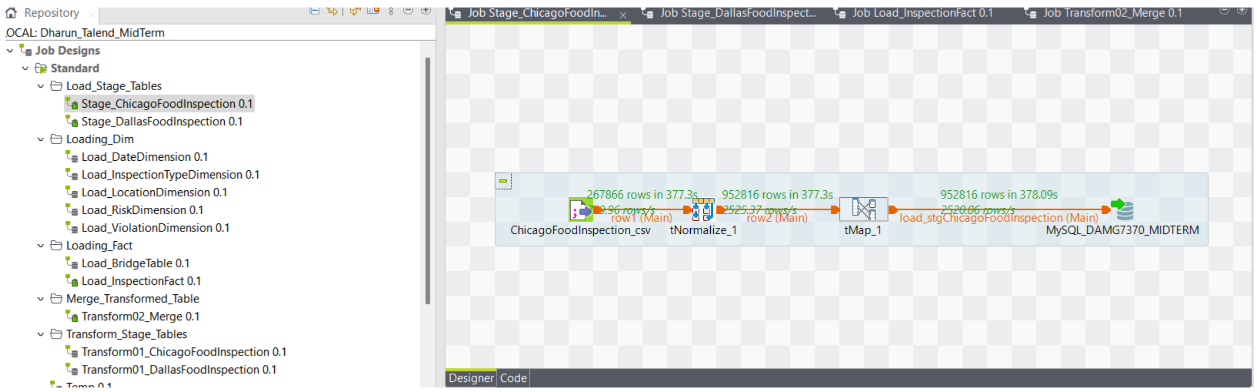
Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra
Address	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
AKA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
City	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DBA_Name	varchar(100)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
DI_CreatedDate	datetime		YES			select,insert,update,references	
DI_WorkFlowFileName	varchar(30)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Facility_Type	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Inspection_Date	datetime		YES			select,insert,update,references	
Inspection_Type	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
InspectionSK	int		YES			select,insert,update,references	
Latitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Longitude	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Results	varchar(50)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Risk	varchar(20)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
State	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Code	varchar(10)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Comment	varchar(6000)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Name	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	
Violation_Point	int		YES			select,insert,update,references	
Zip	varchar(200)		YES	utf8mb4	utf8mb4_0900_...	select,insert,update,references	



LOCAL: Dharun_Talend_MidTerm
- Job Designs
  - Standard
    - Load_Stage_Tables
      - Stage_ChicagoFoodInspection 0.1
      - Stage_DallasFoodInspection 0.1
    - Loading_Dim
    - Loading_Fact
      - Load_BridgeTable 0.1
      - Load_InspectionFact 0.1
    - Merge_Transformed_Table
      - Transform02_Merge 0.1
    - Transform_Stage_Tables
    - Temp 0.1
- Joblet Designs
- Contexts
- Code
- SQL Templates
- Metadata
- Documentation

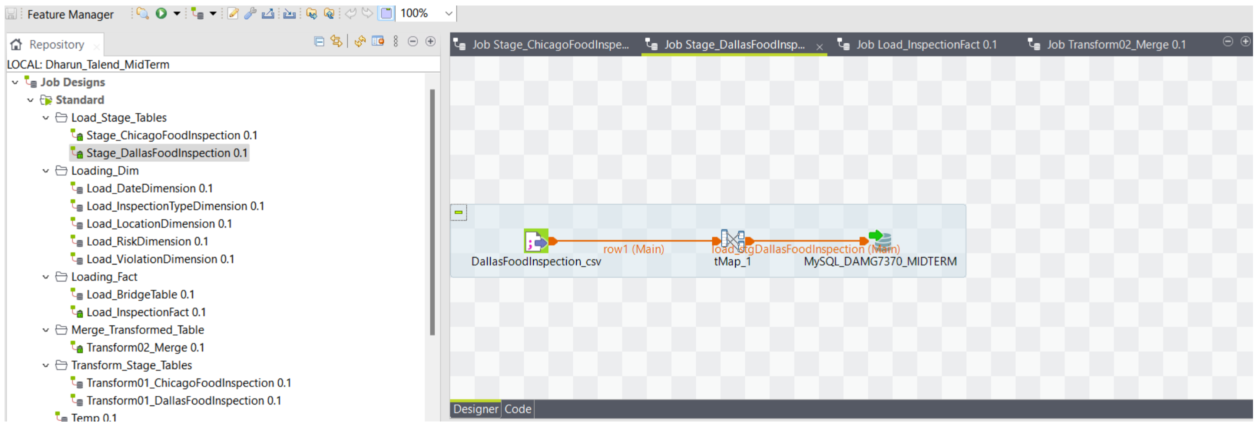
Screenshot of all Talend Jobs

1. Stage_ChicagoFoodInspection



OCAL: Dharun_Talend_MidTerm
- Job Designs
  - Standard
    - Load_Stage_Tables
      - Stage_ChicagoFoodInspection 0.1
      - Stage_DallasFoodInspection 0.1
    - Loading_Dim
      - Load_DateDimension 0.1
      - Load_InspectionTypeDimension 0.1
      - Load_LocationDimension 0.1
      - Load_RiskDimension 0.1
      - Load_ViolationDimension 0.1
    - Loading_Fact
      - Load_BridgeTable 0.1
      - Load_InspectionFact 0.1
    - Merge_Transformed_Table
      - Transform02_Merge 0.1
    - Transform_Stage_Tables
      - Transform01_ChicagoFoodInspection 0.1
      - Transform01_DallasFoodInspection 0.1
    - Temp 0.1

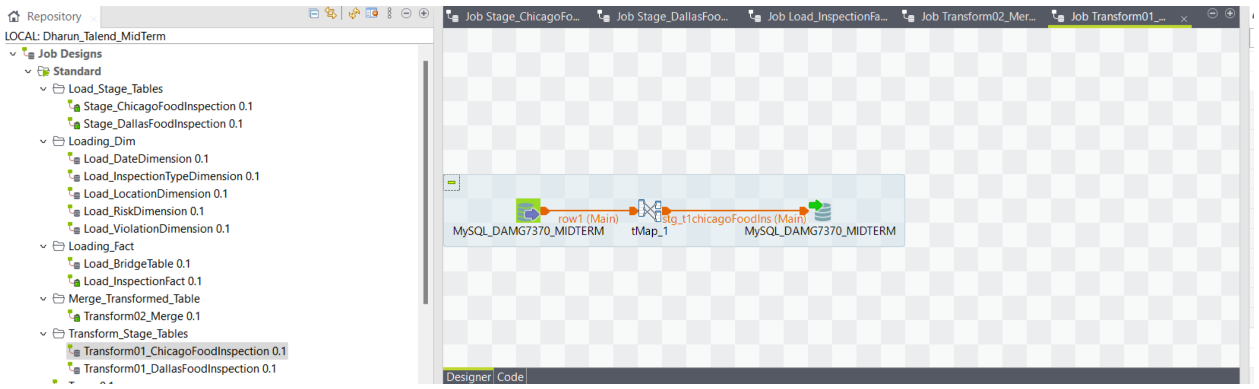
2. Stage_DallasFoodInspection



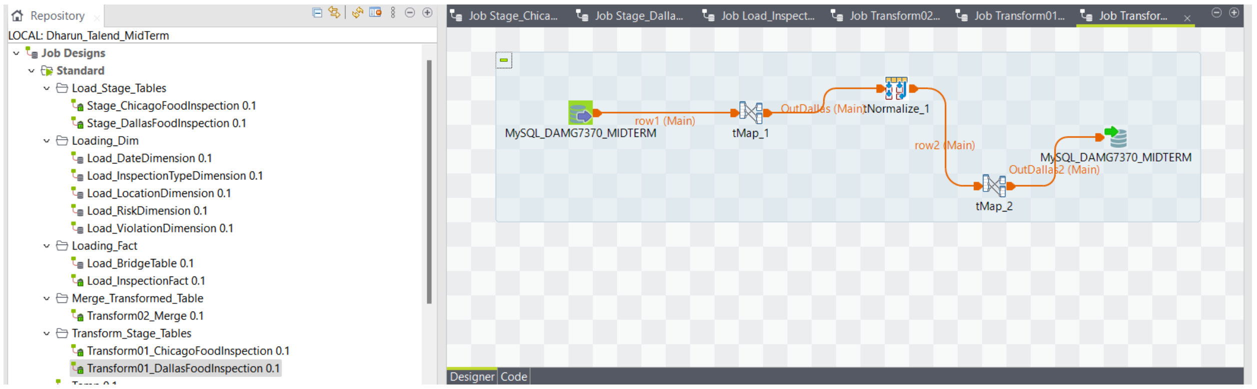
Feature Manager
100%
Repository
Job Stage_ChicagoFoodInspe...
Job Stage_DallasFoodInsp...
Job Load_InspectionFact 0.1
Job Transform02_Merge 0.1
DallasFoodInspection_csv
row1 (Main)
tMap_1
load_stgDallasFoodInspection (Main)
MySQL_DAMG7370_MIDTERM
Designer
Code

LOCAL: Dharun_Talend_MidTerm
- Job Designs
  - Standard
    - Load_Stage_Tables
      - Stage_ChicagoFoodInspection 0.1
      - Stage_DallasFoodInspection 0.1
    - Loading_Dim
      - Load_DateDimension 0.1
      - Load_InspectionTypeDimension 0.1
      - Load_LocationDimension 0.1
      - Load_RiskDimension 0.1
      - Load_ViolationDimension 0.1
    - Loading_Fact
      - Load_BridgeTable 0.1
      - Load_InspectionFact 0.1
    - Merge_Transformed_Table
      - Transform02_Merge 0.1
    - Transform_Stage_Tables
      - Transform01_ChicagoFoodInspection 0.1
      - Transform01_DallasFoodInspection 0.1
    - Temp 0.1

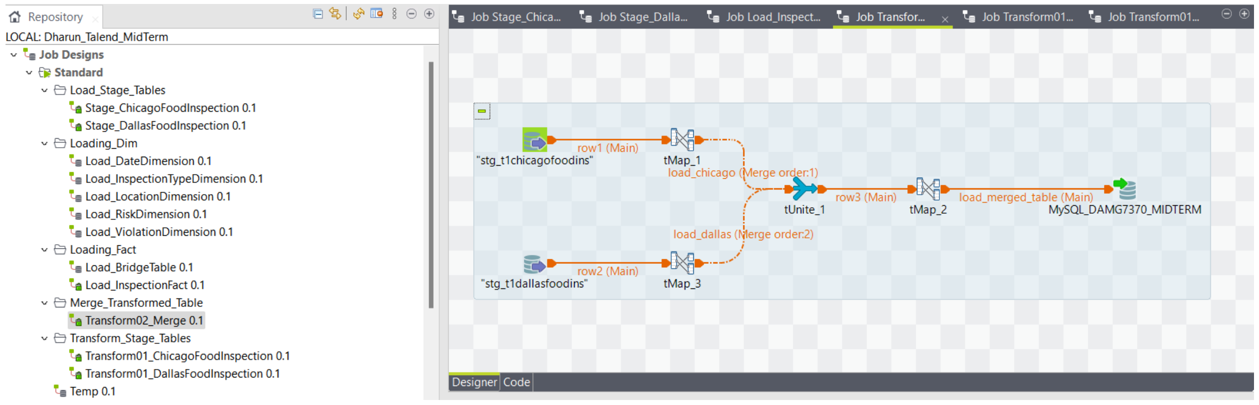
3. Transform01_ChicagoFoodInspection



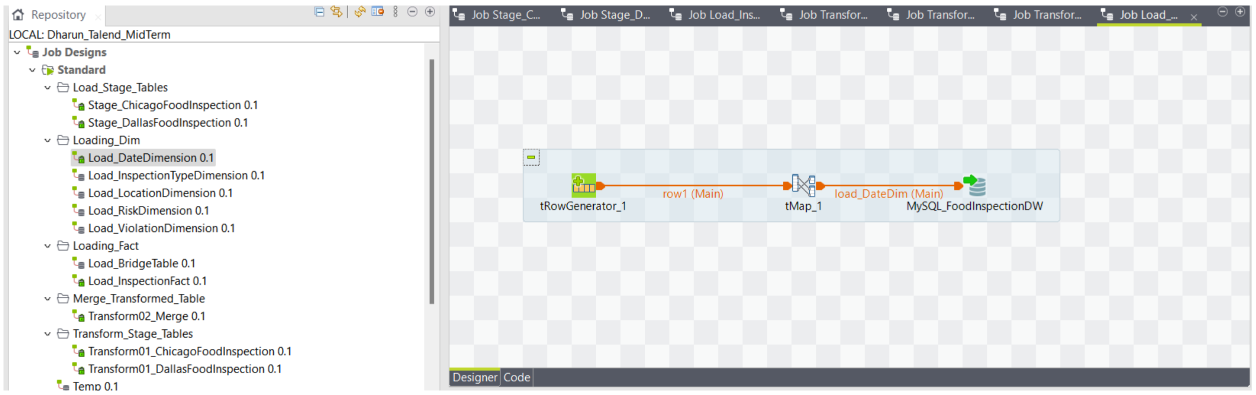
4. Transform01_DallasFoodInspection



5. Transform02_Merge

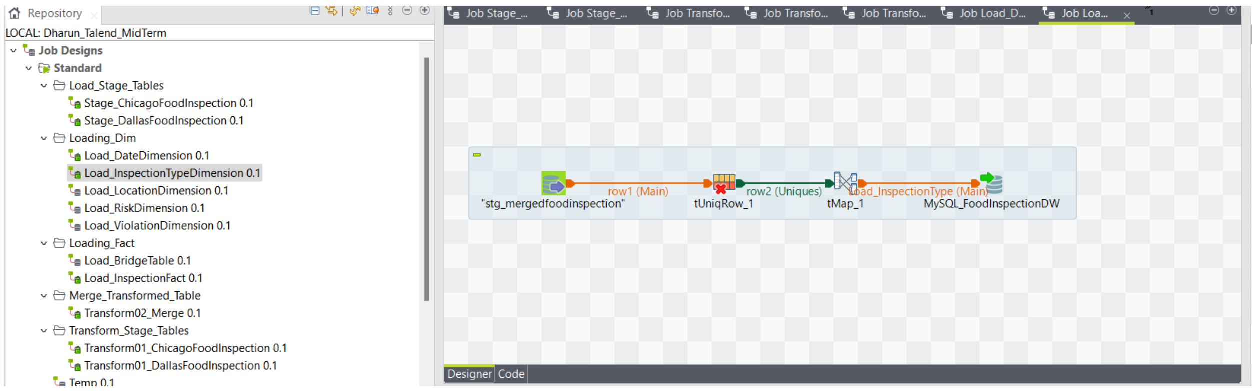


6. Load_DateDimension



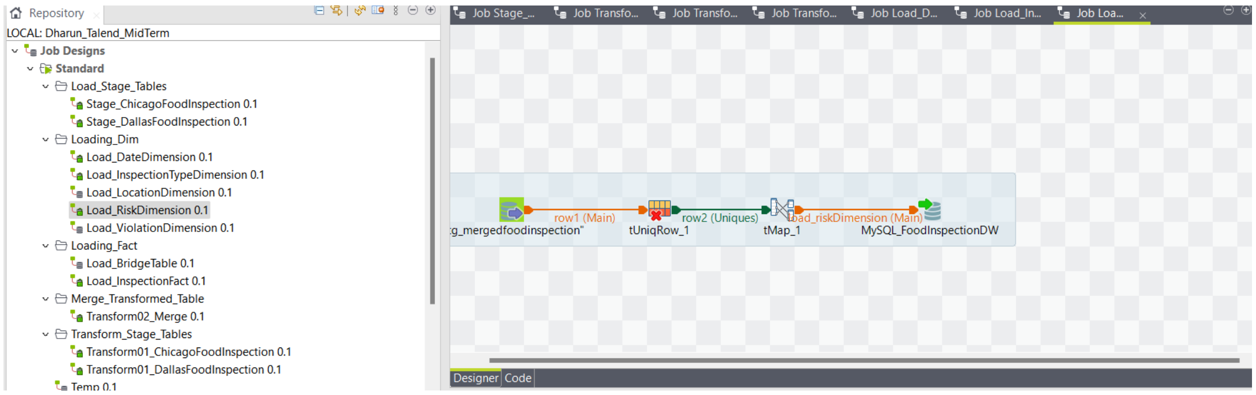
```
LOCAL: Dharun_Talend_MidTerm
v Job Designs
  v Standard
    v Load_Stage_Tables
        Stage_ChicagoFoodInspection 0.1
        Stage_DallasFoodInspection 0.1
    v Loading_Dim
        Load_DateDimension 0.1
        Load_InspectionTypeDimension 0.1
        Load_LocationDimension 0.1
        Load_RiskDimension 0.1
        Load_ViolationDimension 0.1
    v Loading_Fact
        Load_BridgeTable 0.1
        Load_InspectionFact 0.1
    v Merge_Transformed_Table
        Transform02_Merge 0.1
    v Transform_Stage_Tables
        Transform01_ChicagoFoodInspection 0.1
        Transform01_DallasFoodInspection 0.1
    Temp 0.1
```

7. Load_InspectionTypeDimension



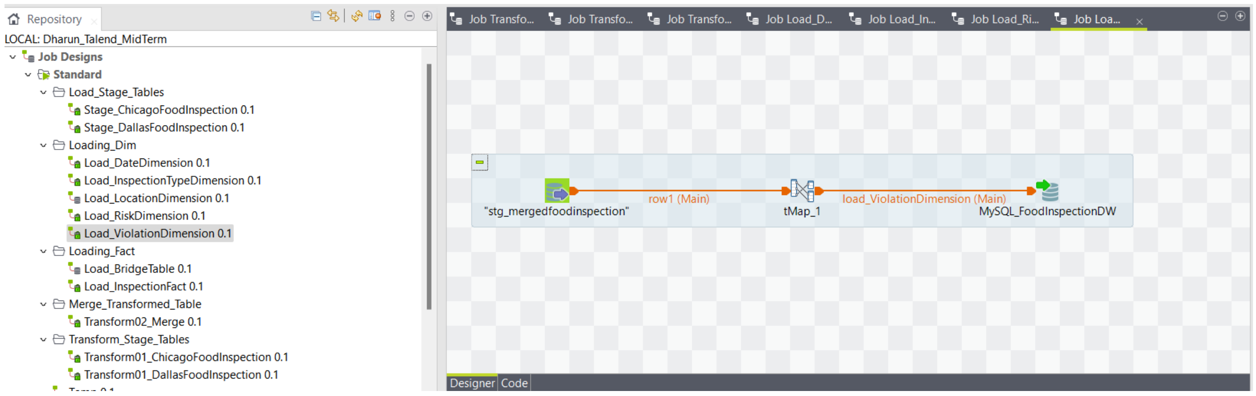
```
LOCAL: Dharun_Talend_MidTerm
v Job Designs
  v Standard
    v Load_Stage_Tables
        Stage_ChicagoFoodInspection 0.1
        Stage_DallasFoodInspection 0.1
    v Loading_Dim
        Load_DateDimension 0.1
        Load_InspectionTypeDimension 0.1
        Load_LocationDimension 0.1
        Load_RiskDimension 0.1
        Load_ViolationDimension 0.1
    v Loading_Fact
        Load_BridgeTable 0.1
        Load_InspectionFact 0.1
    v Merge_Transformed_Table
        Transform02_Merge 0.1
    v Transform_Stage_Tables
        Transform01_ChicagoFoodInspection 0.1
        Transform01_DallasFoodInspection 0.1
    Temp 0.1
```

8. Load_RiskDimension

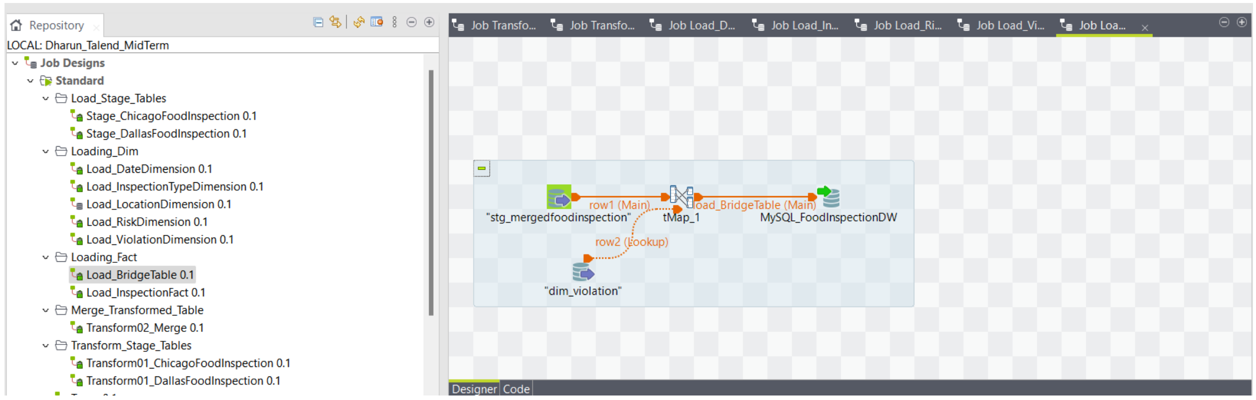


```
LOCAL: Dharun_Talend_MidTerm
v Job Designs
  v Standard
    v Load_Stage_Tables
        Stage_ChicagoFoodInspection 0.1
        Stage_DallasFoodInspection 0.1
    v Loading_Dim
        Load_DateDimension 0.1
        Load_InspectionTypeDimension 0.1
        Load_LocationDimension 0.1
        Load_RiskDimension 0.1
        Load_ViolationDimension 0.1
    v Loading_Fact
        Load_BridgeTable 0.1
        Load_InspectionFact 0.1
    v Merge_Transformed_Table
        Transform02_Merge 0.1
    v Transform_Stage_Tables
        Transform01_ChicagoFoodInspection 0.1
        Transform01_DallasFoodInspection 0.1
    Temp 0.1
```

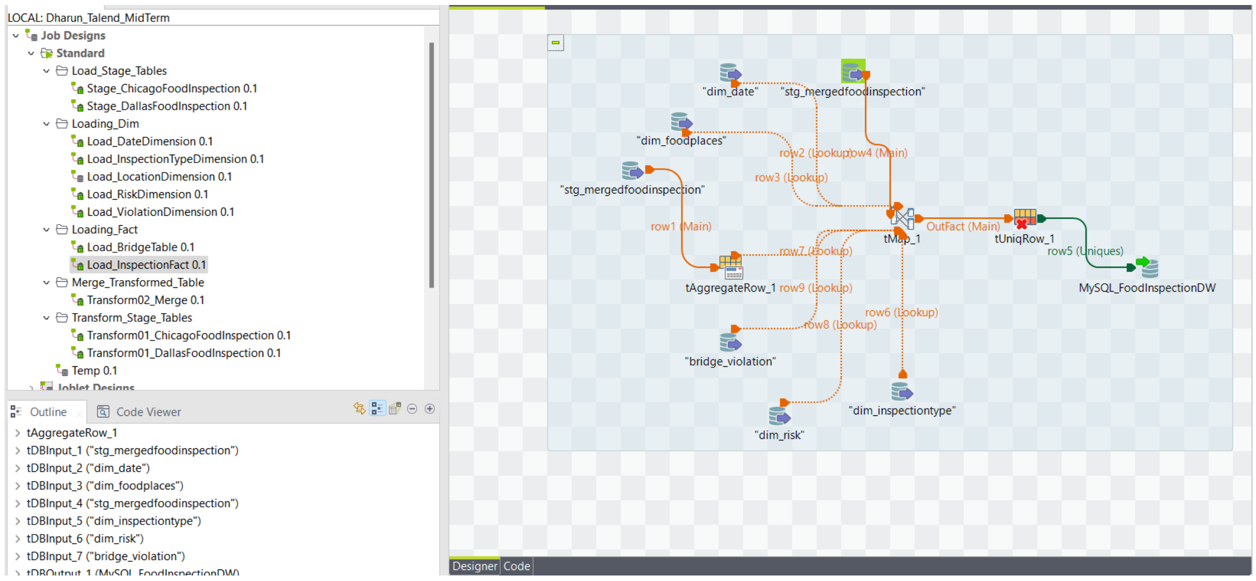
9. Load_ViolationDimension



10. Load_BridgeTable

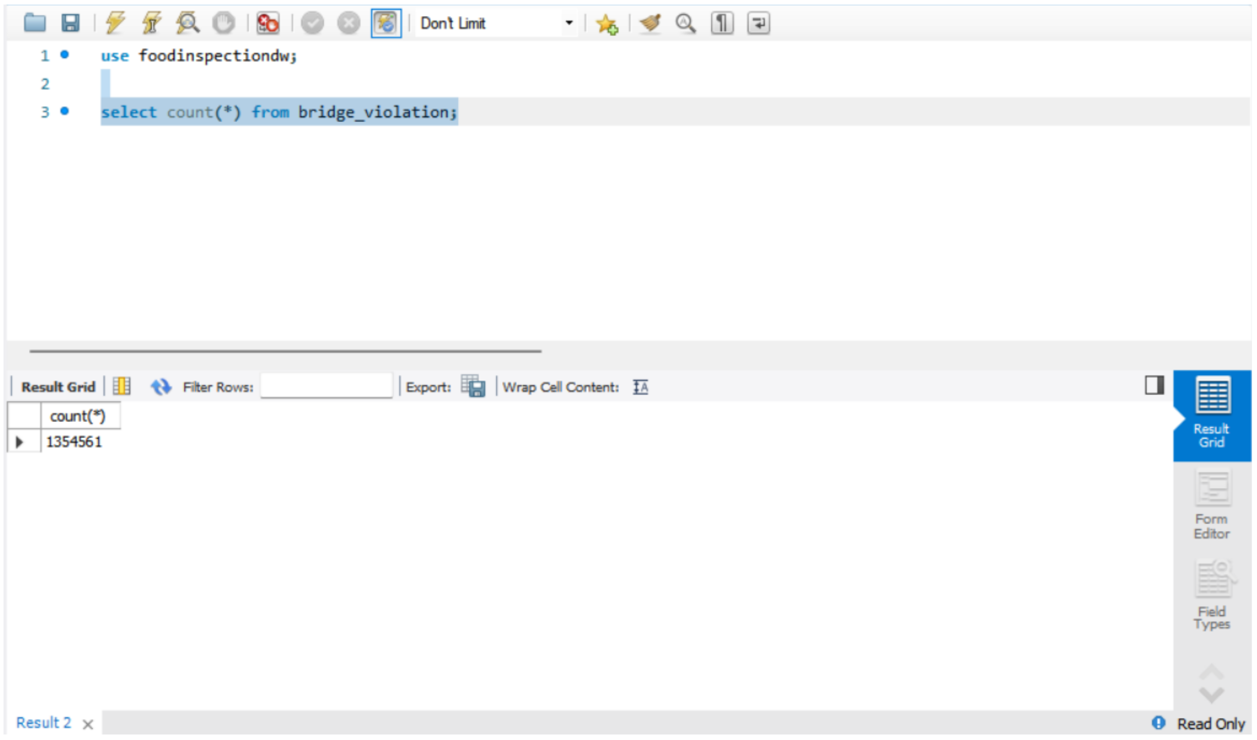


11. Load_InspectionFact



Target Table Row Counts

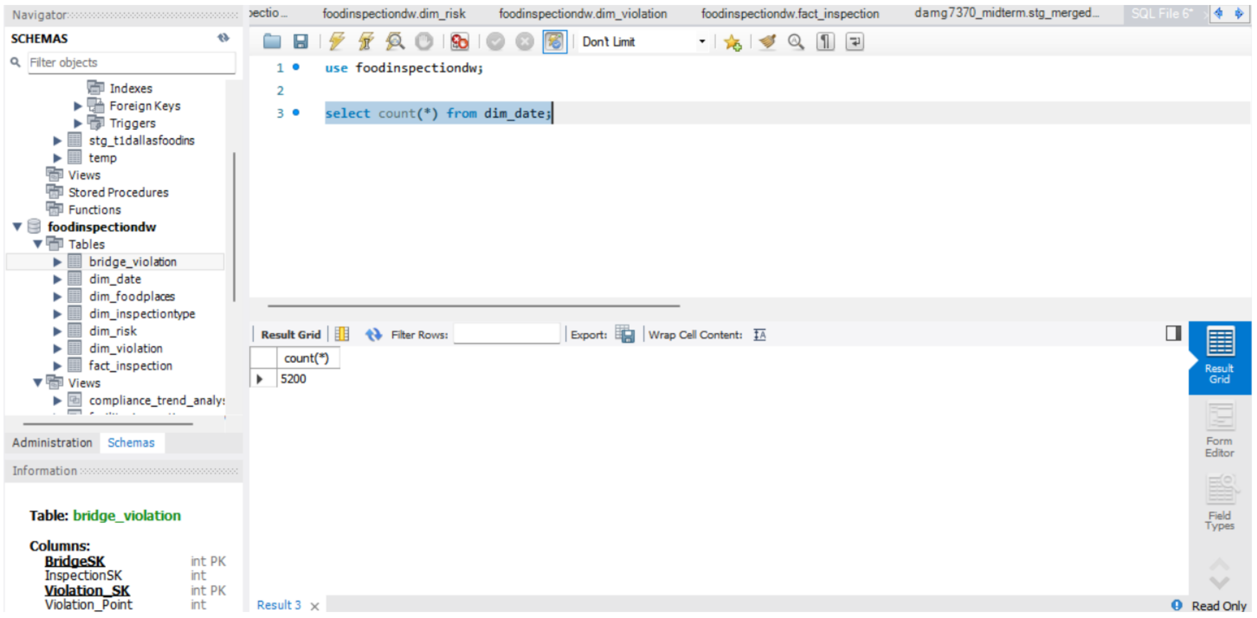
1. Bridge Violations



```
use foodinspectiondw;

select count(*) from bridge_violation;
```

```
count(*)
1354561
```

2. DateDimension



```
use foodinspectiondw;

select count(*) from dim_date;
```

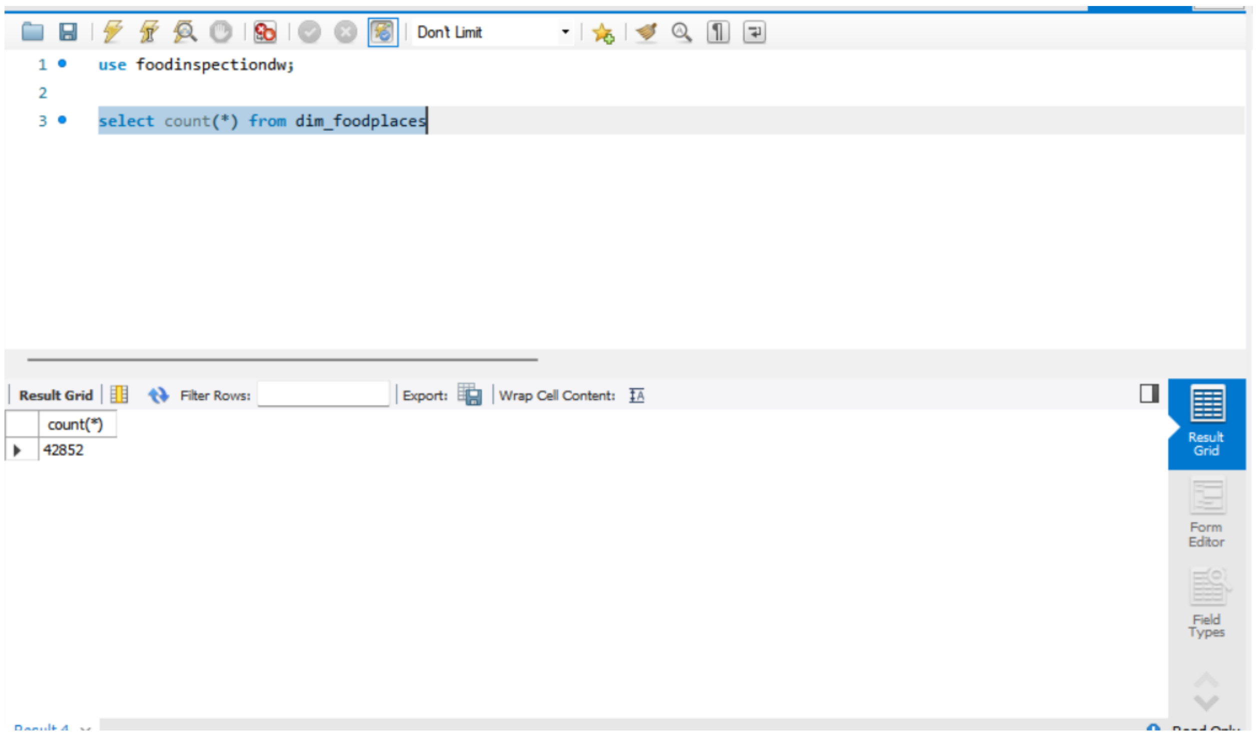
```
count(*)
5200
```

Table: bridge_violation

Columns:

BridgeSK	int PK
InspectionSK	int
Violation_SK	int PK
Violation_Point	int

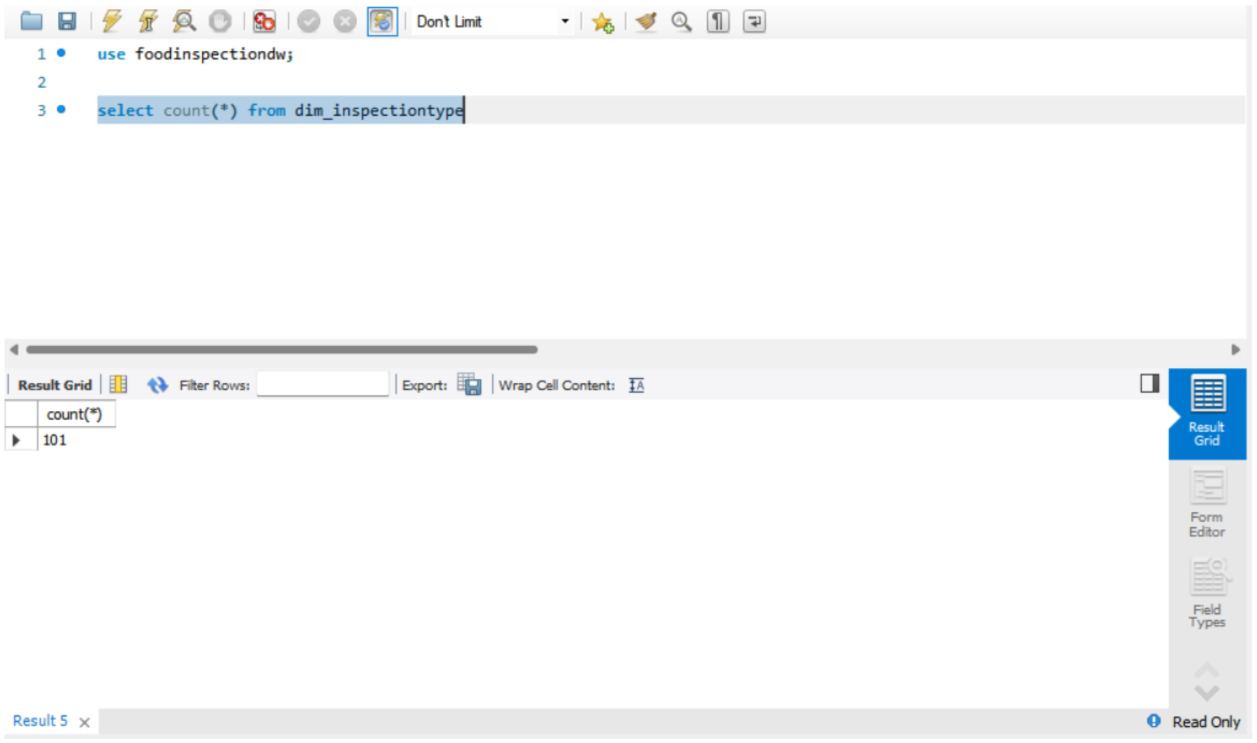
3. FoodPlacesDimension



```
use foodinspectiondw;

select count(*) from dim_foodplaces
```

```
count(*)
42852
```

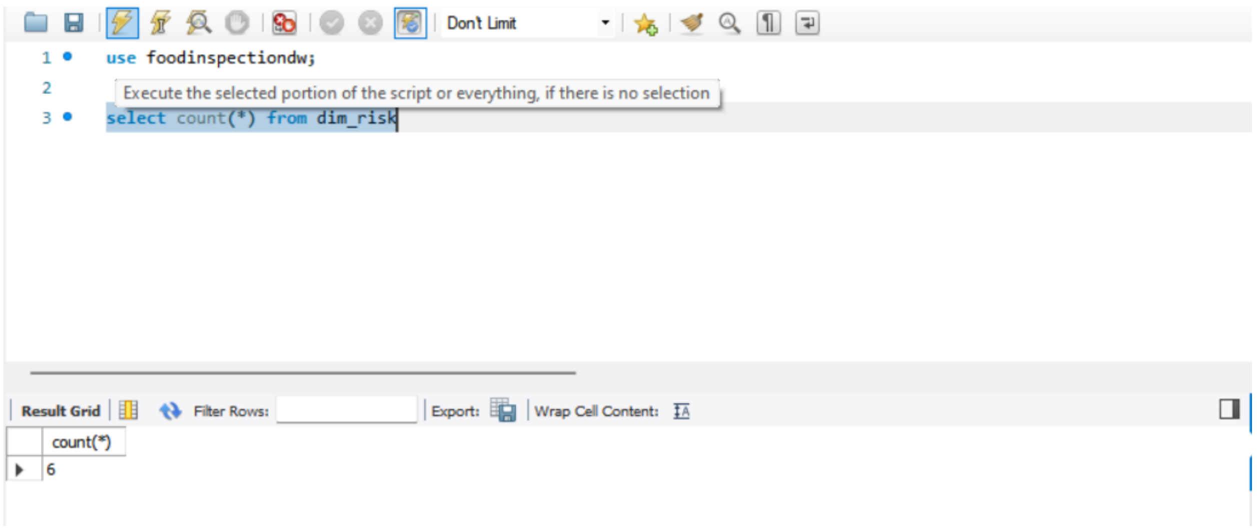
4. InspectionTypeDimension



```
use foodinspectiondw;

select count(*) from dim_inspectiontype
```

```
count(*)
101
```

5. RiskDimension



Don't Limit

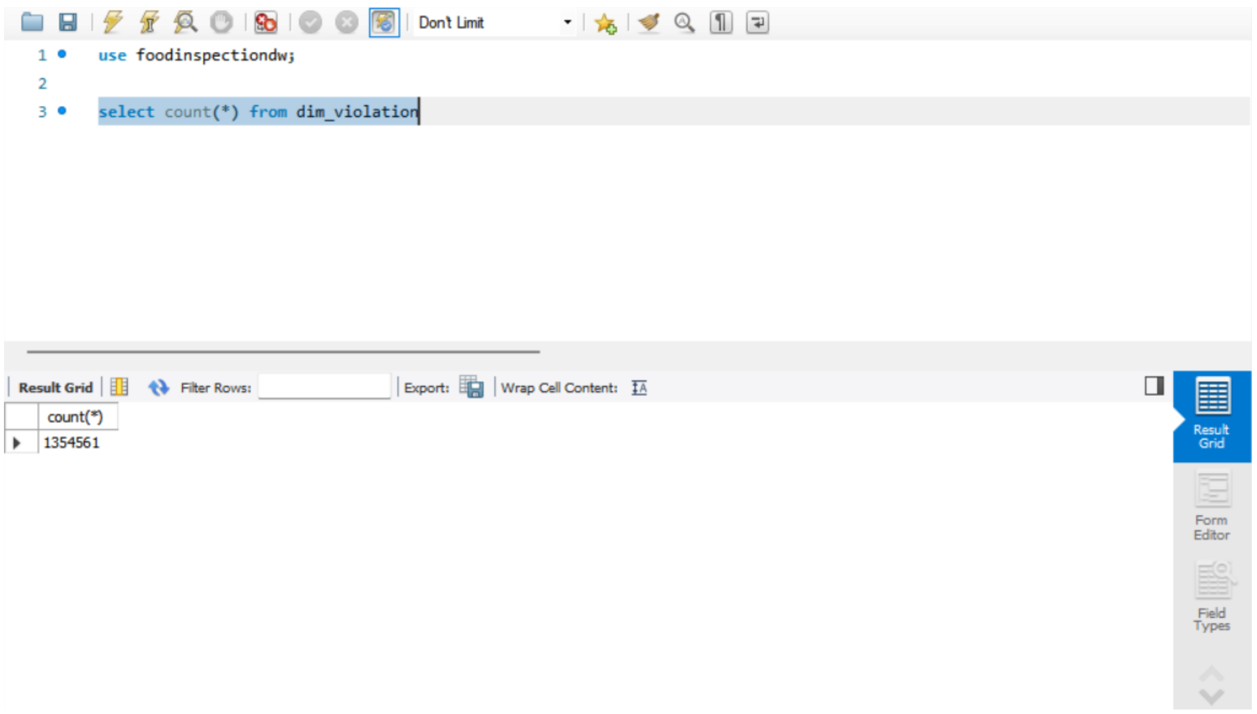
```
use foodinspectiondw;

select count(*) from dim_risk
```

Execute the selected portion of the script or everything, if there is no selection

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
count(*)
6
```

6. **ViolationDimension**



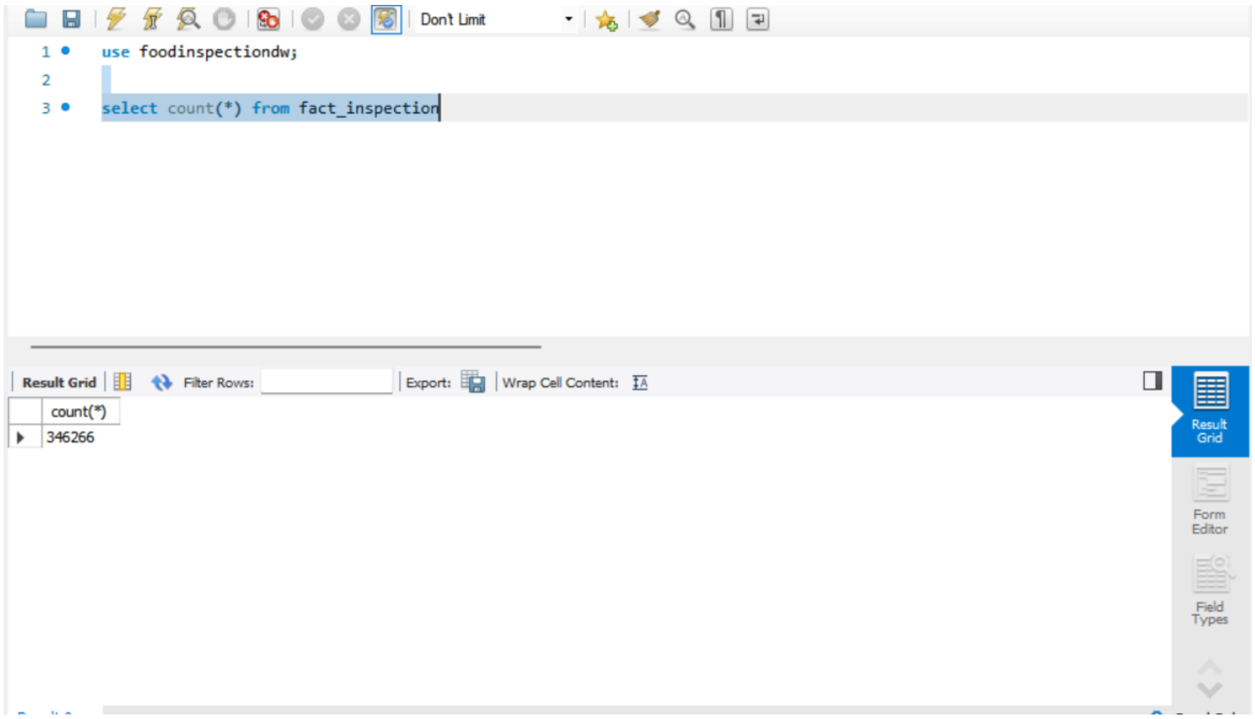
Don't Limit

```
use foodinspectiondw;

select count(*) from dim_violation
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

```
count(*)
1354561
```

7. **InspectionFact**



```
use foodinspectiondw;

select count(*) from fact_inspection
```

```
count(*)
346266
```