Food Inspections (Dallas and Chicago)

After performing Data Profiling on both the datasets: We saw for missing values, inconsistent values and data types to match up with the dimension tables:

First we made data dictionary separately for Dallas and Chicago datasets, we listed down column names and data types that were coming from staging tables so it was easier while loading into dim and facts

Dallas Food Inspection : Data Catalog

Restaurant Name	Varchar(65) Null	
Inspection Type	Char(9) Null	Code indicating the inspection type, such as **Routine, Follow-up, Complaint, Temporary and Mobile.** • Routine Inspections – are conducted at least once every six months • Follow-up Inspections – are conducted as a result of poor sanitation issues, low scores • Complaints Inspections – General Sanitation/Hygienic Practices /Illness Investigation, Smoking and Other • Temporary – the City of Dallas Office of Special Events provides a listing of public events being held involving food and the Consumer Health Division provides guidance and inspects • Mobile – the various mobile food units are inspected annually with random inspections conducted during the year
Inspection_Date	Date	
Inspection score	Smallint	The aggregate score from the inspection violations. Please note not every violation will reflect a point deduction as establishments are allowed to correct violations during the inspection process, and therefore no reduction in the overall score is reflected for the violation

Street Number	Int null	Eg 123
Street Name	Varchar(25)	Eg Name
Street Direction	Char(1)	Eg N, E
Street Type	Char(4)	Eg LD
Street Unit	Char(5)	Eg Unit of shop
Zip Code	Char(10)	Eg
Violation Description		(Violation Number + Description)
Violation points		
Violation Detail		
Inspection Month	Char(8)	
Inspection FY	Char(6)	
Latitude	Float	
Longitude	Float	

<u>Chicago Food Inspection : Data Catalog</u>

Inspection ID, DBA Name, AKA Name, License #, Facility Type, Risk, Address, City, State, Zip, Inspection Date, Inspection Type, Results, Latitude, Longitude, Location, Inspection_Date, Violations, Violation Code, Violation Description, Violation Comments

Business_License

ID	Char(16)	
LICENSE ID	Int	An internal database ID for each record. Each license can have multiple records as it goes through renewals and other transactions. See the LICENSE NUMBER field for the number generally known to the public and used in most other data sources that refer to the license. License ID is used from Business_License Dataset of Chicago to connect with FoodInspection Dataset of Chicago

FoodInspection_Chiacgo

Inspection ID	Int	
DBA Name	Varchar(79)	
AKA Name	Varchar(79)	
License #	Int	
Facility Type	Varchar(47)	
Risk	Char(15)	
Address	Varchar(51)	

City	Varchar(20)	
State	char()	
Zip	int	
Inspection Date	date	
Inspection Type	varchar()	
Results	varchar()	
Latitude	char()	
Longitude	char()	
Inspection_Date	date	
Location	char()	
Violations	varchar()	
Violation Code	int	
Violation Description	varchar()	

Violation Comments	varchar()	

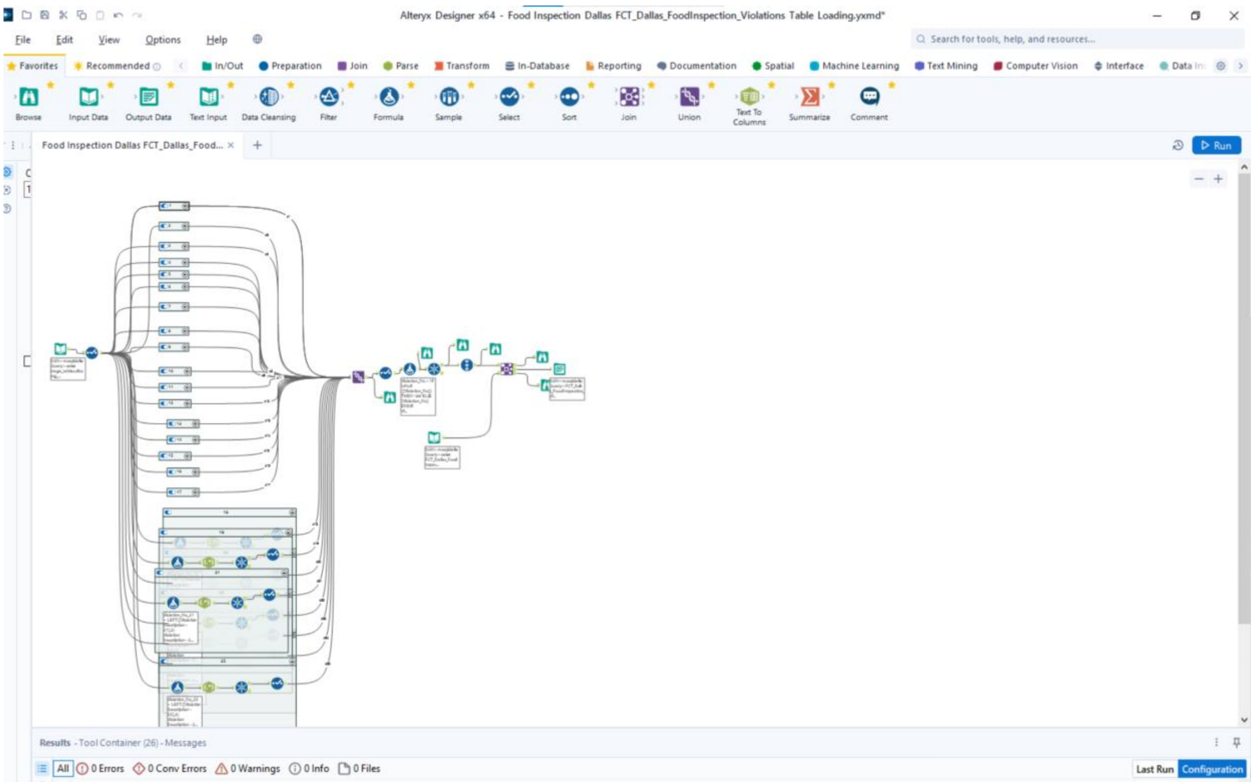
General Insights for Chicago and Dallas Dataset:

1) In Dallas, had to Spli the lat long location column into latitude and longitude using regex, data cleansing, text to column connectors

2) In Dallas, Since the violation attributes in Dallas had it type of data that is in the wide format(there were 25 violation attributes), we took the 25 unique combination of Violation Description -1, Violation Points -1, Violation Detail-1, Violation Memo -1, similarly for 2 till 25 and then we performed union to poulate this data into a single common column for each of the violation attributes.

Violation Description - 1
Violation Points - 1
Violation Detail - 1
Violation Memo - 1
Violation Description - 2
Violation Points - 2
Violation Detail - 2
Violation Memo - 2
Violation Description - 3
Violation Points - 3
Violation Detail - 3
Violation Memo - 3
Violation Description - 4
Violation Points - 4
Violation Detail - 4
Violation Memo - 4
Violation Description - 5
Violation Points - 5
Violation Detail - 5
Violation Memo - 5
Violation Description - 6
Violation Points - 6
Violation Detail - 6
Violation Memo - 6

3) <u>Fact 2 Loading:</u>



4) In dallas we found that the address attributes like the column Street Address was a combination of (Street Number, Street Name, Street Type, Street Direction, Street Unit). So initially while loading the food inspection fact we joined on all this attributes from staging tables with the dim dallas geo. But after studying the dataset we found some discrepancies like here you can see for the first row: the combination works but for the second row although

# Street Number street_number	T Street Name street_name	T Street Direction street_direction	T Street Type street_type	T Street Unit street_unit	T Street Address site_address	T Zip Code zip
1328	JIM MILLER	N	RD	#104	1328 N JIM MILLER RD #104	75217

Discrepancies example: here although the street unit is null, still the street address shows some kind of value for street unit as #334. So when we join with all the address attributes the valid addresses were not being captured in the inner join output due this descrepancies, so we decided to join on only street address and zipcode which are up to date to populate the fact table to get the correct row count.

# Street Number street_number	T Street Name street_name	T Street Direction street_direction	T Street Type street_type	T Street Unit street_unit	T Street Address site_address	T Zip Code zip
9310	FOREST		LN		9310 FOREST LN #334	75243-4217

5) We also found that the number before the description had some standard meaning to it which is actually a violation code which we used further in our combined food inspection.

Violation Descriptio… *violation1_description*
*14 When to wash hands af…
*10 Other sanitizer exposur…
*31 Individual, disposable t…

6) In chicago, we had to split the into latitude and longitude using regex, data cleansing, text to column connectors

7) In chicago, the violation column also had to be split into 3 columns: Violation code, violation description and violation comments, we did that by using regex:

8) We also found that DBA corresponds to legal name, so we joined according to the business document standard definition

Title: Food Inspections

Brief Description: This dataset contains information from inspections of restaurants and other food establishments in Chicago from January 1, 2010 to the present.

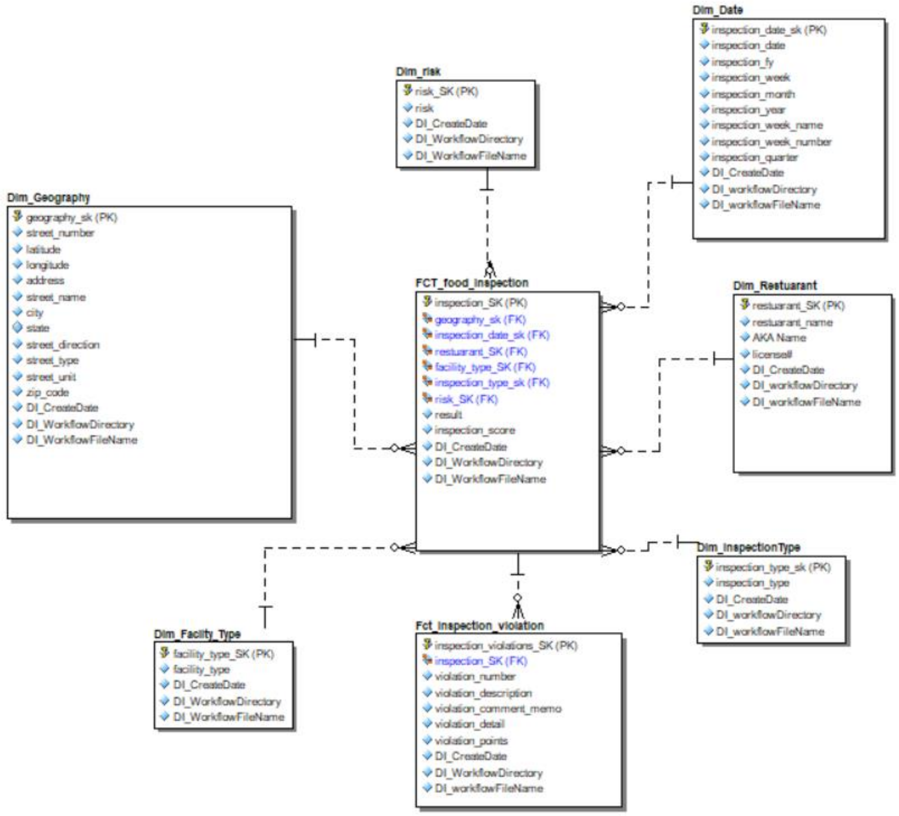
Description: This information is derived from inspections of restaurants and other food establishments in Chicago from January 1, 2010 to the present. Inspections are performed by staff from the Chicago Department of Public Health's Food Protection Program. Inspections are done using a standardized procedure. The results of the inspection are inputted into a database, then reviewed and approved by a State of Illinois Licensed Environmental Health Practitioner (LEHP). A subset of data elements are extracted from this database and downloaded into this data portal. These elements are:

- DBA: 'Doing business as.' This is legal name of the establishment.
- AKA: 'Also known as.' This is the name the public would know the establishment as.
- License number: This is a unique number assigned to the establishment for the purposes of licensing by the Department of Business Affairs and Consumer Protection.

9) Here we joined Chicago inspections dataset and business licenses on the license number, restaurant name and address to get inner joined output as for this all the restaurants that had inspections had license number available.

   Dim Restaurant Reload: Now there were many restaurants in Chicago who had the inspections but did not have the license number attributes present in the business license dataset, so it was not getting captured in the inner join output. For this we performed union on the inner join output and left join output to get all the records that had food inspections recorded for chicago.

<u>Combined Dimensional Model:</u> This is the combined dimensional model that we designed.



We the insights we got from the individual chicago and dallas dataset, used same cleaning process and techniques while loading the dim and facts

Reasons why we choose to make these dimensions:

After analyzing the dataset provided for both the Dallas and Chicago food inspections, we have identified some common entities and attributes that can be modeled in a single data model.

Dim_Restaurant: This entity will contain information about the restaurants where the inspections are being conducted.

Inspection Type: This entity will contain information about the type of inspections conducted.

Dim_Date: This entity will contain inspection dates along with different grain elements that are useful for analyzing the trends (weekly, quarterly etc.)

Dim_Geography: This entity will contain the address attributes associated with the restaurants present in both data set.

Reason for Dim_risk and Dim_Facility_Type as a separate dimension:

This is because in the future there is a possibility of the expansion of these values and it will be better to make changes in the dimension rather than making changes in fact. (Currently this attributes has values only for chicago food dataset)

For the inspection_violation_fact:
We studied the dataset and found that some of the violation attributes can be combined together for both chicago and dallas, (color coded are combined together to represent one column for both dallas and chicago)

Violation Description (Dallas) = (Violation Number + Description)

Violation points (Dallas)

Violation Detail (Dallas)

Violation Memo(Dallas)

Violations(Chicago): which we splitted into the 3 columns belows

Violation Code (Chicago)

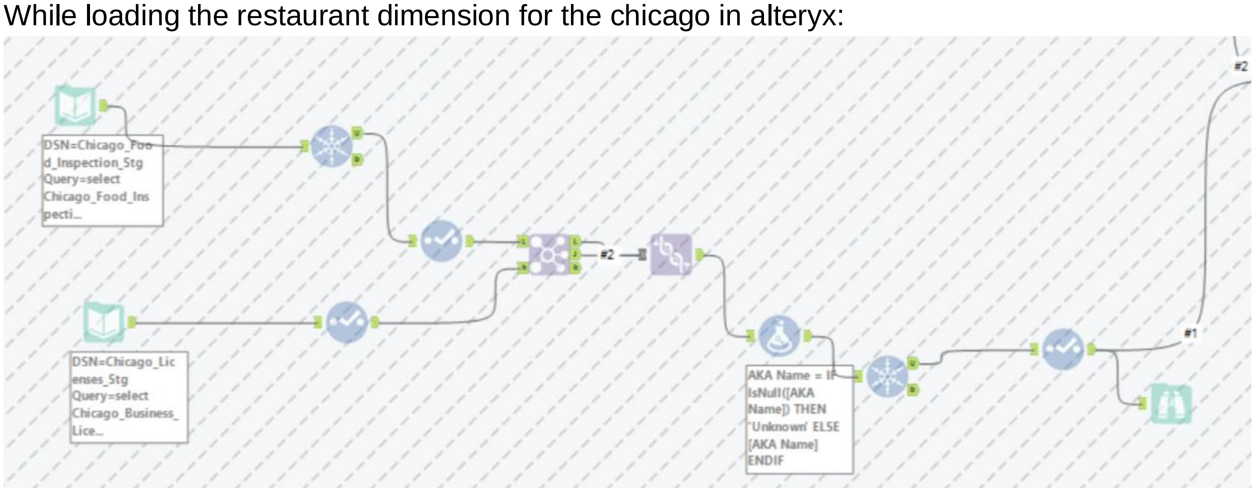
Violation Description(Chicago)

Violation Comments(Chicago)

Reason for excluding the business licenses attributes that are present for the Chicago:
Since the business licenses staging table had licenses for all the entities(food, hospital, school etc), we though of eliminating the unwanted one and only storing the required one.

In dim_restaurant, along with restaurant_name and aka_name, we took the only license number from the business licenses staging. Here also the license number was only available for chicago so had to define default values for the license number attribute as -9999 for dallas records since we were combining it into a common dim.

While loading the restaurant dimension for the chicago in alteryx:



DISCREPANCIES:

There are some discrepancies that were observed while loading the data into the common data model:

Data Types: Some of the attributes in the original data sets had distinct data types. The Inspection Date attribute, for instance, was of type datetime in the Dallas data set while it was of type date in the Chicago data set. In the shared data model, we had to make sure that the attribute data types were consistent between the two data collections.

Attribute Names: The attribute names in the original data sets were not consistent. For example, in the Dallas dataset there was an Inspection Score attribute, whereas in the Chicago dataset there were Results.

Attributes like City and State were missing from the Dallas dataset, hence we imputation them with default values, City - Dallas & State - TX.

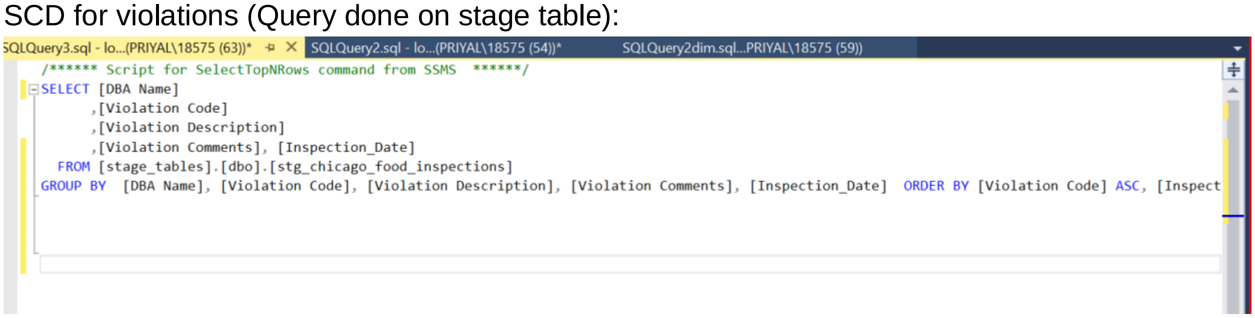
Similarly, there were other attributes that were missing in the Dallas dataset but were essential to the data model, hence we imputed them with default values. We had to ensure that the attribute names in the common data model were uniform across both data sets.

Primary Keys: The primary keys in the original data sets were not consistent. For example, the primary key for the FCT_Inspection table in the Dallas data set was named InspectionSK, whereas the primary key for the FCT_Inspection table in the Chicago data set was named InspectionID. We had to make sure that the primary keys were consistent across both data sets in the common data model.

Data Quality: We observed some issues with the data quality in both data sets. For example, some of the restaurants had missing or incorrect address information. We had to perform some data cleaning and validation to ensure the quality of the data before loading it into the common data model.

SCD:

SCD for violations (Query done on stage table):

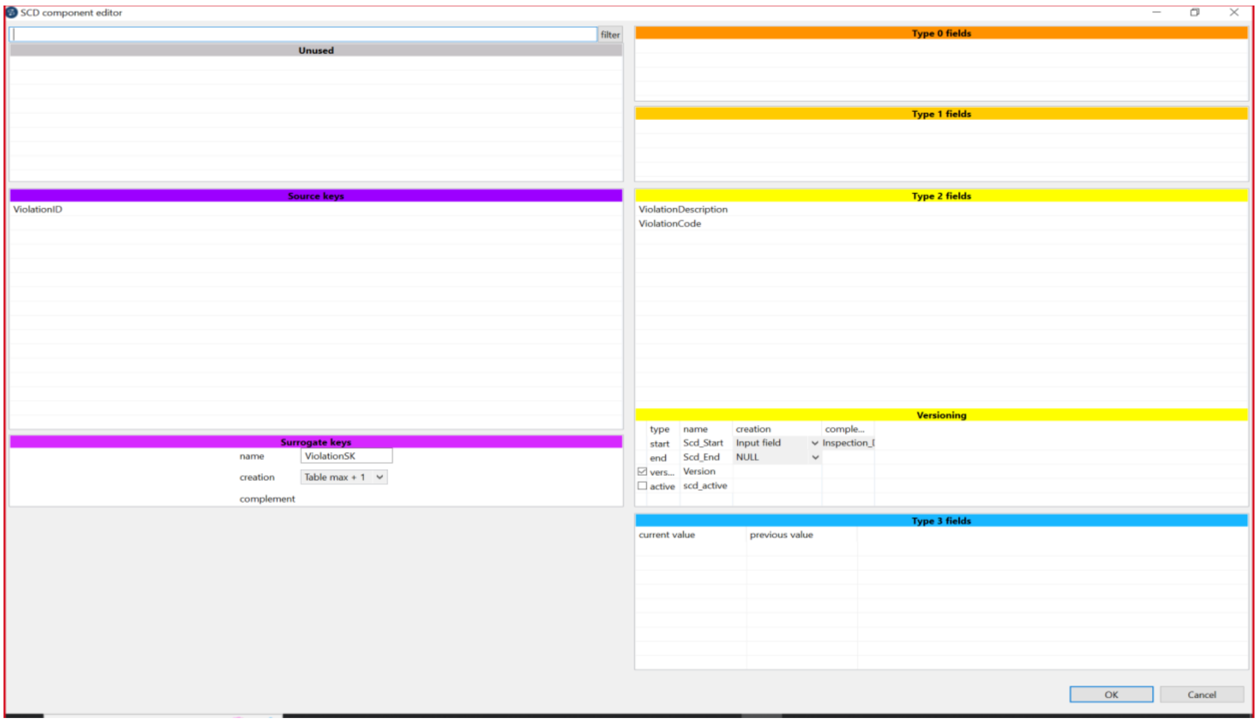


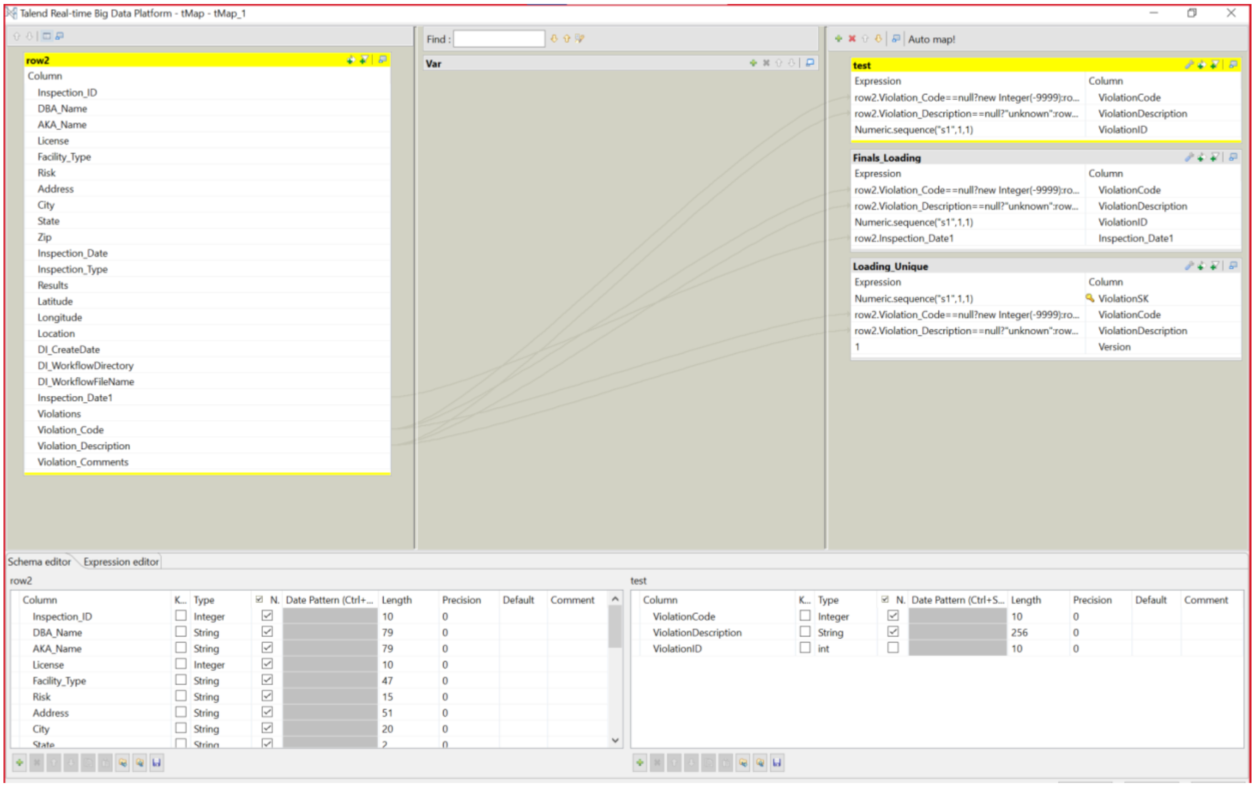

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT [DBA Name]
      ,[Violation Code]
      ,[Violation Description]
      ,[Violation Comments], [Inspection_Date]
  FROM [stage_tables].[dbo].[stg_chicago_food_inspections]
GROUP BY  [DBA Name], [Violation Code], [Violation Description], [Violation Comments], [Inspection_Date]  ORDER BY [Violation Code] ASC, [Inspect
```

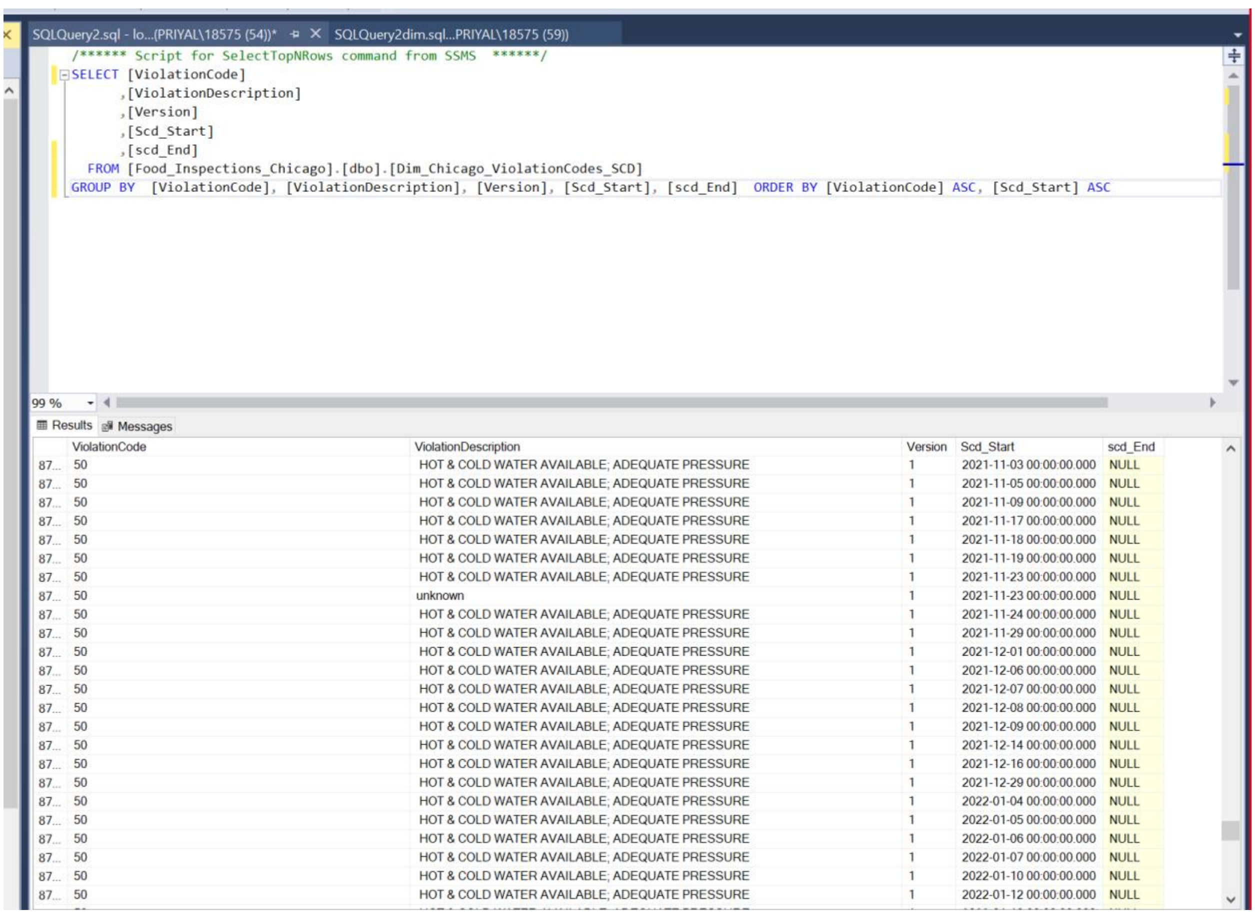
	DBA Name	Violation Code	Violation Description	Violation Comments	Inspection_Date
7	#1 CHINA EXPRESS, LTD.	NULL	NULL	NULL	2012-09-14
8	#1 CHOP SUEY	33	FOOD AND NON-FOOD CONTACT EQUIPMENT UTENSILS CLEA...	Can opener, storage shelves, exhaust hood and filters not cleaned. I...	2010-06-28
9	#1 CHOP SUEY	34	FLOORS: CONSTRUCTED PER CODE, CLEANED, GOOD REPAIR...	Floor under equipment and along wallbase not cleaned. Instructed t...	2010-06-28
10	#1 CHOP SUEY	35	WALLS, CEILINGS, ATTACHED EQUIPMENT CONSTRUCTED PE...	Ceiling tile in prep area not maintained. Instructed to detail all damag...	2010-06-28
11	#1 CHOP SUEY	2	FACILITIES TO MAINTAIN PROPER TEMPERATURE	REACH IN PREP COOLER IN POOR REPAIR, INTERNAL TEMPE...	2010-07-01
12	#1 CHOP SUEY	35	WALLS, CEILINGS, ATTACHED EQUIPMENT CONSTRUCTED PE...	INTERIOR OF HOOD AND LIGHTSHIELDS THRU-OUT NOT CLE...	2010-07-01
13	#1 CHOP SUEY	3	POTENTIALLY HAZARDOUS FOOD MEETS TEMPERATURE REQ...	THE FOLLOWING POTENTIALLY HAZARDOUS FOODS AT IMPR...	2010-07-01
14	#1 CHOP SUEY	31	CLEAN MULTI-USE UTENSILS AND SINGLE SERVICE ARTICLES...	INSTD TO STORE CLEAN KNIVES PROPERLY.	2010-07-01
15	#1 CHOP SUEY	38	VENTILATION: ROOMS AND EQUIPMENT VENTED AS REQUIRE...	PIPE LEAKING UNDER PREP SINK. REPAIR SAME.	2010-07-01
16	#1 CHOP SUEY	35	WALLS, CEILINGS, ATTACHED EQUIPMENT CONSTRUCTED PE...	The walls and ceilings shall be in good repair and easily cleaned. TH...	2010-07-06
17	#1 CHOP SUEY	NULL	NULL	NULL	2010-07-12
18	#1 CHOP SUEY	33	FOOD AND NON-FOOD CONTACT EQUIPMENT UTENSILS CLEA...	CLEAN THE FOLLOWING:INTERIOR CABINETS AND EXTERIOR...	2011-01-05
19	#1 CHOP SUEY	33	FOOD AND NON-FOOD CONTACT EQUIPMENT UTENSILS CLEA...	All food and non-food contact surfaces of equipment and all food st...	2011-01-05
20	#1 CHOP SUEY	30	FOOD IN ORIGINAL CONTAINER, PROPERLY LABELED: CUSTO...	MUST LABEL AND DATE ALL COOKED AND PREPARED FOOD...	2011-01-05

Query executed successfully. | DESKTOP-NVKRNPT (15.0 RTM) | DESKTOP-NVKRNPT\panch ... | stg_Chicago_Food_Inspe... | 00:00:13 | 879,593 rows

The scd table we implemented using Prof Rick Sherman’s data model:



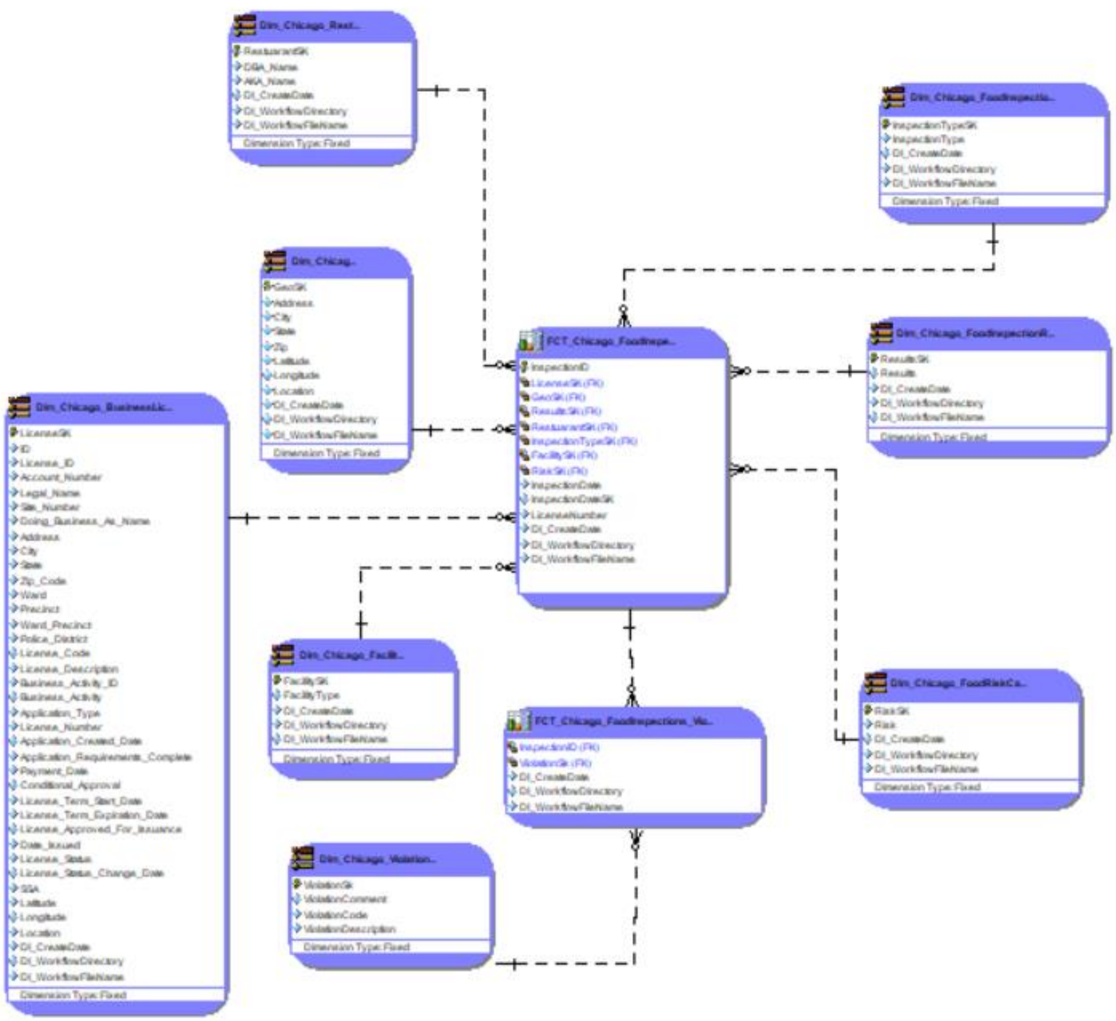

Talend Real-time Big Data Platform - tMap - tMap_1
row2
Column
Inspection_ID
DBA_Name
AKA_Name
License
Facility_Type
Risk
Address
City
State
Zip
Inspection_Date
Inspection_Type
Results
Latitude
Longitude
Location
DI_CreateDate
DI_WorkflowDirectory
DI_WorkflowFileName
Inspection_Date1
Violations
Violation_Code
Violation_Description
Violation_Comments
Find :
Var
Auto map!
test
Expression
Column
row2.Violation_Code==null?new Integer(-9999):ro...
ViolationCode
row2.Violation_Description==null?"unknown":row...
ViolationDescription
Numeric.sequence("s1",1,1)
ViolationID
Finals_Loading
Expression
Column
row2.Violation_Code==null?new Integer(-9999):ro...
ViolationCode
row2.Violation_Description==null?"unknown":row...
ViolationDescription
Numeric.sequence("s1",1,1)
ViolationID
row2.Inspection_Date1
Inspection_Date1
Loading_Unique
Expression
Column
Numeric.sequence("s1",1,1)
ViolationSK
row2.Violation_Code==null?new Integer(-9999):ro...
ViolationCode
row2.Violation_Description==null?"unknown":row...
ViolationDescription
1
Version
Schema editor
Expression editor
row2
Column | K... | Type | N. | Date Pattern (Ctrl+... | Length | Precision | Default | Comment
Inspection_ID | Integer | 10 | 0
DBA_Name | String | 79 | 0
AKA_Name | String | 79 | 0
License | Integer | 10 | 0
Facility_Type | String | 47 | 0
Risk | String | 15 | 0
Address | String | 51 | 0
City | String | 20 | 0
test
Column | K... | Type | N. | Date Pattern (Ctrl+S... | Length | Precision | Default | Comment
ViolationCode | Integer | 10 | 0
ViolationDescription | String | 256 | 0
ViolationID | int | 10 | 0


SQLQuery2.sql - lo...(PRIYAL\18575 (54))*
SQLQuery2dim.sql...PRIYAL\18575 (59))
/****** Script for SelectTopNRows command from SSMS ******/
SELECT [ViolationCode]
,[ViolationDescription]
,[Version]
,[Scd_Start]
,[scd_End]
FROM [Food_Inspections_Chicago].[dbo].[Dim_Chicago_ViolationCodes_SCD]
GROUP BY [ViolationCode], [ViolationDescription], [Version], [Scd_Start], [scd_End] ORDER BY [ViolationCode] ASC, [Scd_Start] ASC
99 %
Results
Messages
ViolationCode | ViolationDescription | Version | Scd_Start | scd_End
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-03 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-05 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-09 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-17 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-18 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-19 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-23 00:00:00.000 | NULL
87... 50 | unknown | 1 | 2021-11-23 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-24 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-11-29 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-12-01 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-12-06 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-12-07 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-12-08 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-12-09 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-12-14 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-12-16 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2021-12-29 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2022-01-04 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2022-01-05 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2022-01-06 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2022-01-07 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2022-01-10 00:00:00.000 | NULL
87... 50 | HOT & COLD WATER AVAILABLE; ADEQUATE PRESSURE | 1 | 2022-01-12 00:00:00.000 | NULL

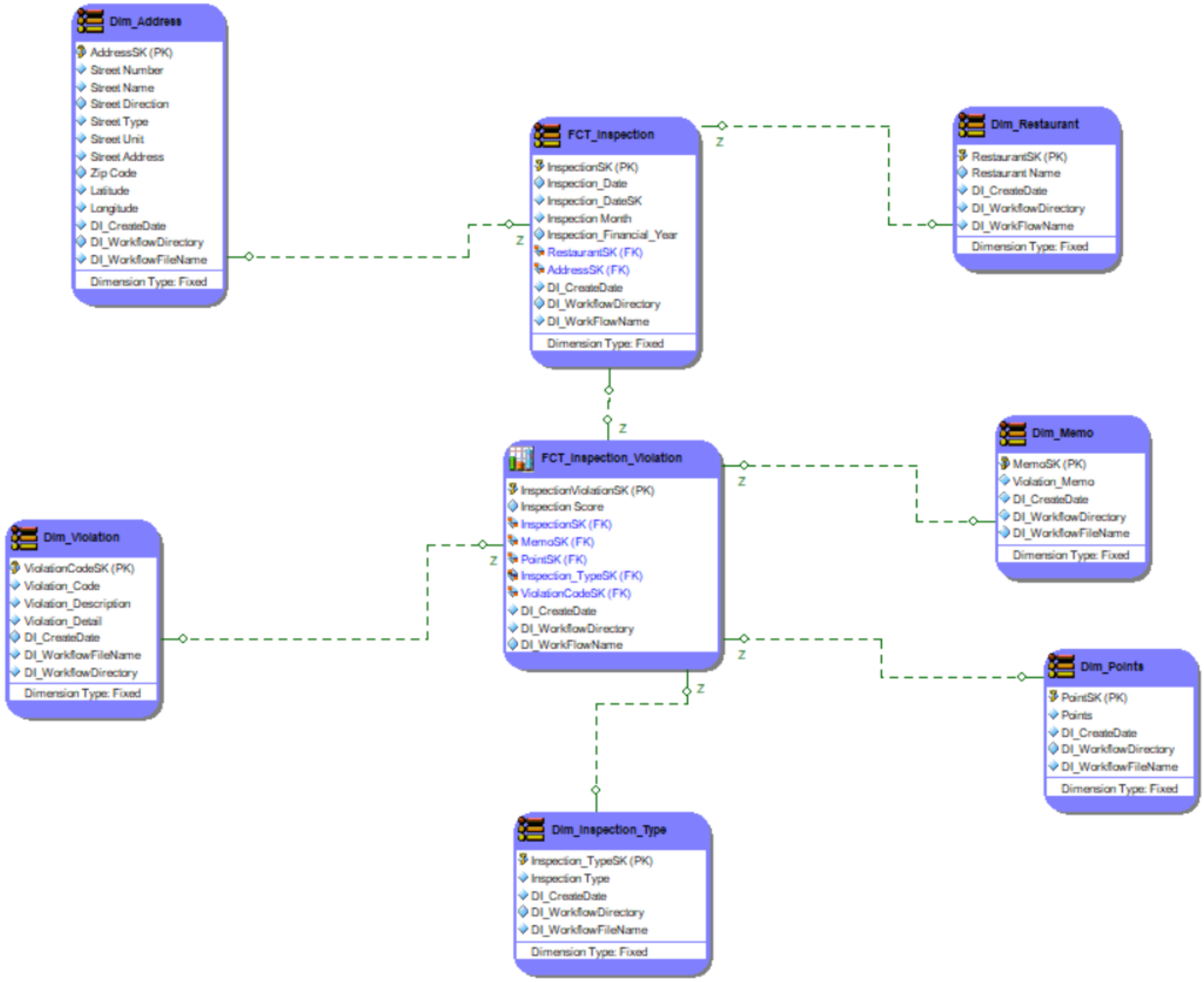
Previously, In the data model which was provided to us there was an scd for Violation_Code as in the above SS we can see that for a DBA Name there is a Violation code change at every inspection_date. But there was an error in the Loading of the SCD in data model as in the first SS attach we can see that the Version column is coming out to be same for all the rows and as there was no end date provided to us in the data set and in the data model the End_date column is NULL for all the rows.

Now, we have not Shown SCD in our version of the data model because in the original dataset of chicago we just have inspection date but no where the end_date is mentioned of the inspection date in the dataset. So, if we want to implement scd correctly then first we have to track all the inspection dates of the DBA and make a new column where we will store when the inspection is ended. This column will help us populate the End_date column in the scd and also gives us the correct version.

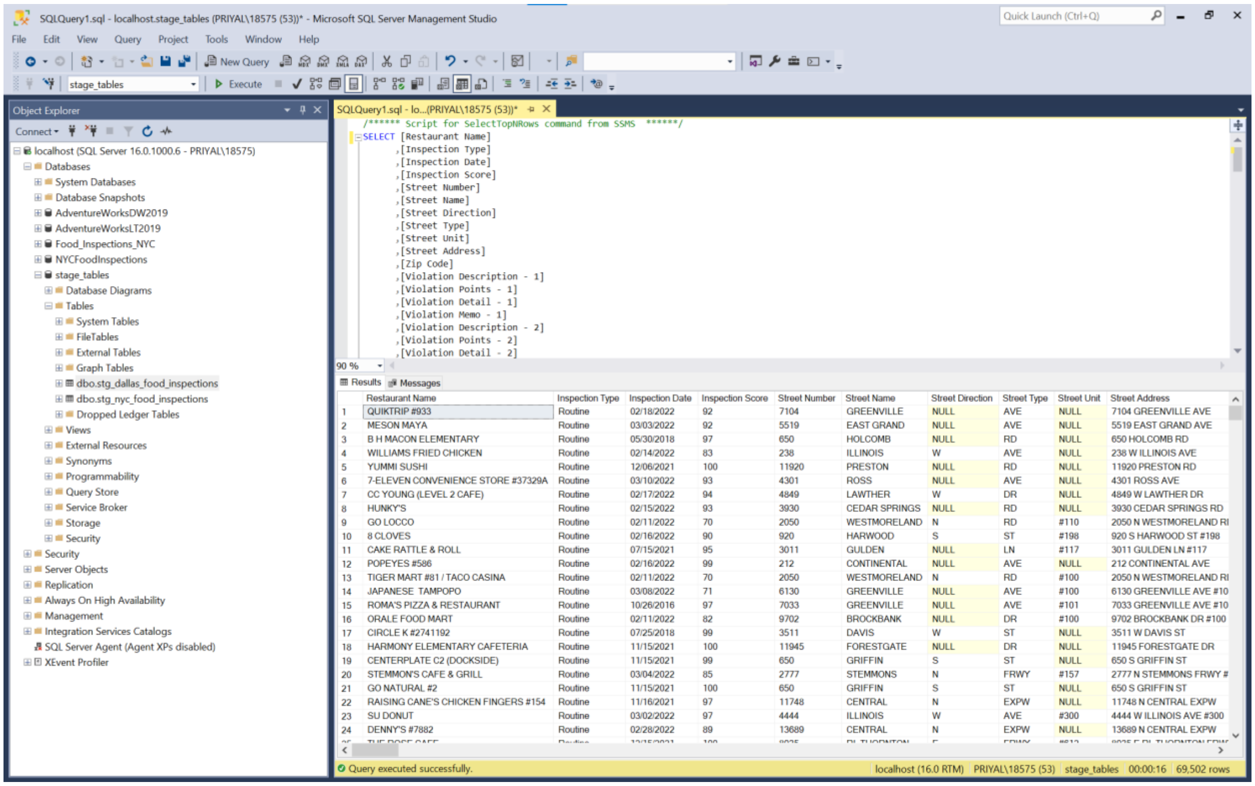
<u>Chicago Dimensional Model:</u>



Dallas Dimensional Model:



Dallas food inspections staging row count: 69502



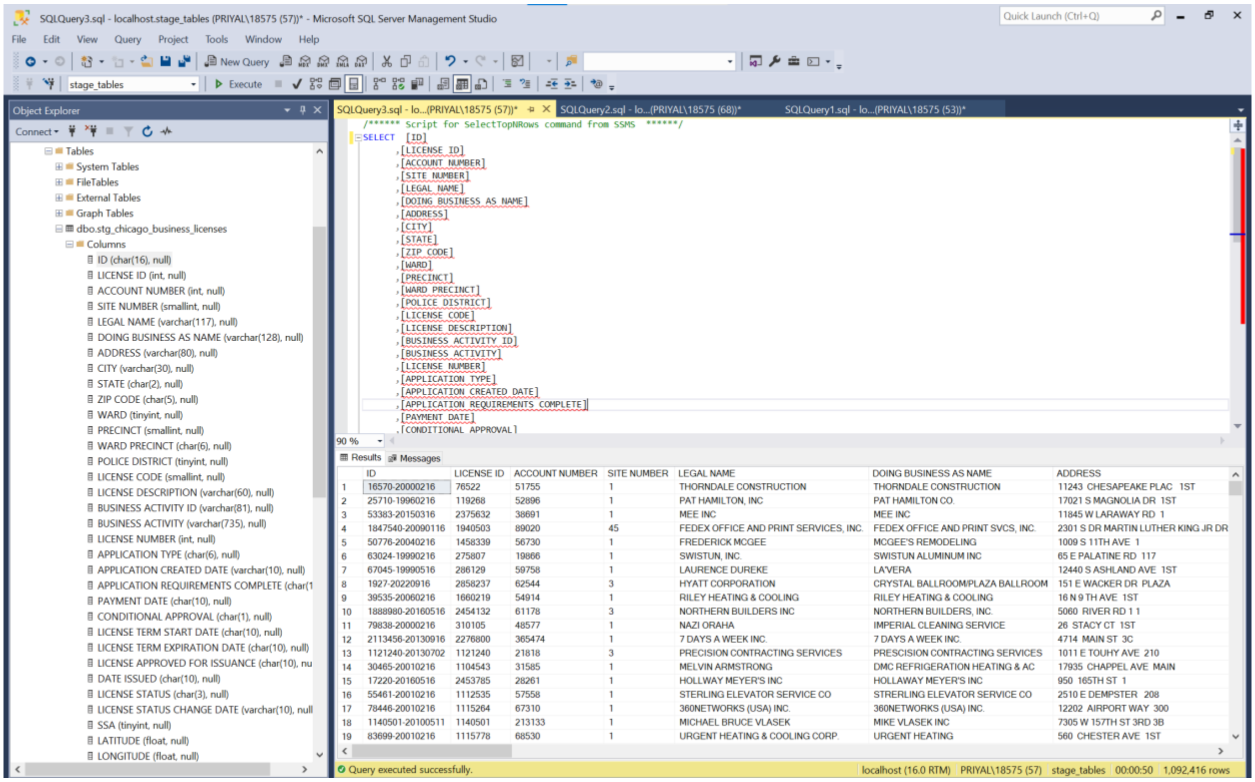
Dallas Dimensional model row count:

99 %

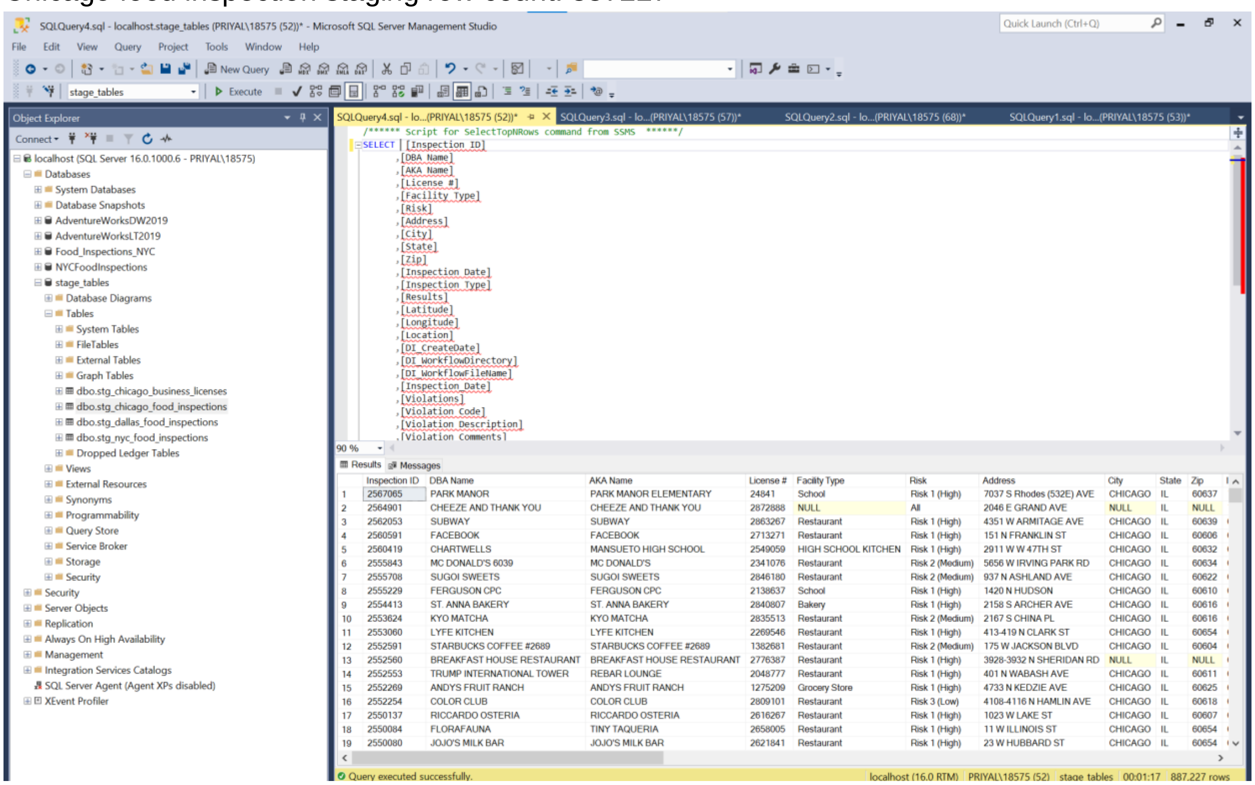
Results | Messages

	name	row_count
1	Dim_Dallas_Geo	7687
2	Dim_Dallas_InspectionType	3
3	Dim_Dallas_Restaurant	9949
4	FCT_Dallas_FoodInspection	10594
5	FCT_Dallas_FoodInspection_Violations	88200

Chicago business licenses staging row count: 1092416



Chicago food inspection staging row count: 887227



Chicago Dimensional model row count:

	name	row_count
1	Dim_Chicago_BusinessLicenses	1094853
2	Dim_Chicago_FacilityType	456
3	Dim_Chicago_FoodInspectionResults	7
4	Dim_Chicago_FoodInspectionType	98
5	Dim_Chicago_FoodRiskCategory	5
6	Dim_Chicago_Geo	19197
7	Dim_Chicago_Restaurants	32517
8	Dim_Chicago_ViolationCodes_SCD	99367
9	FCT_Chicago_FoodInspections	85690
10	FCT_Chicago_FoodInspections_Viol...	836010

Combined Dimensional model row count:

Results Messages

	TableName	RecordCount
1	dbo.Dim_Date	7410
2	dbo.Dim_Facilty_Type	912
3	dbo.Dim_Geography	53782
4	dbo.Dim_InspectionType	202
5	dbo.Dim_Restuarant	105374
6	dbo.Dim_risk	5
7	dbo.FCT_food_inspection	95254
8	dbo.Fct_inspection_violation	894450