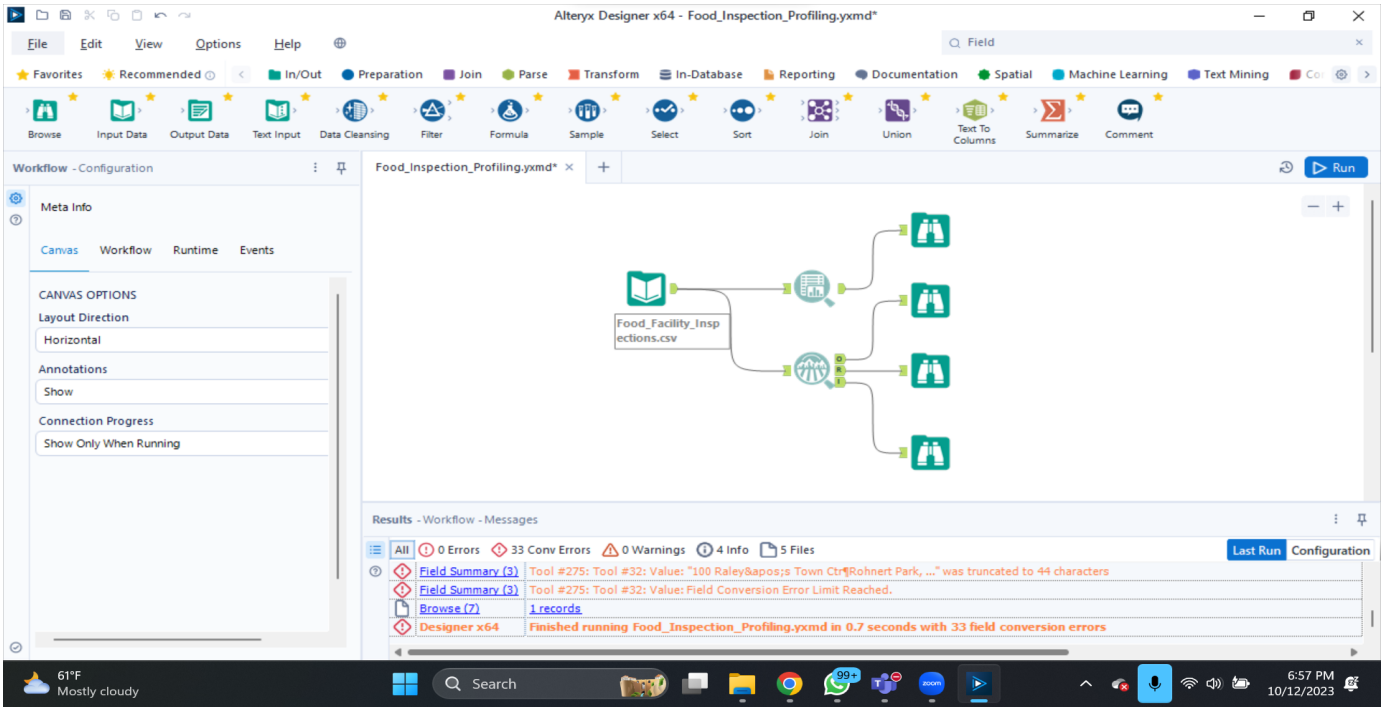
Food Inspection Documentation

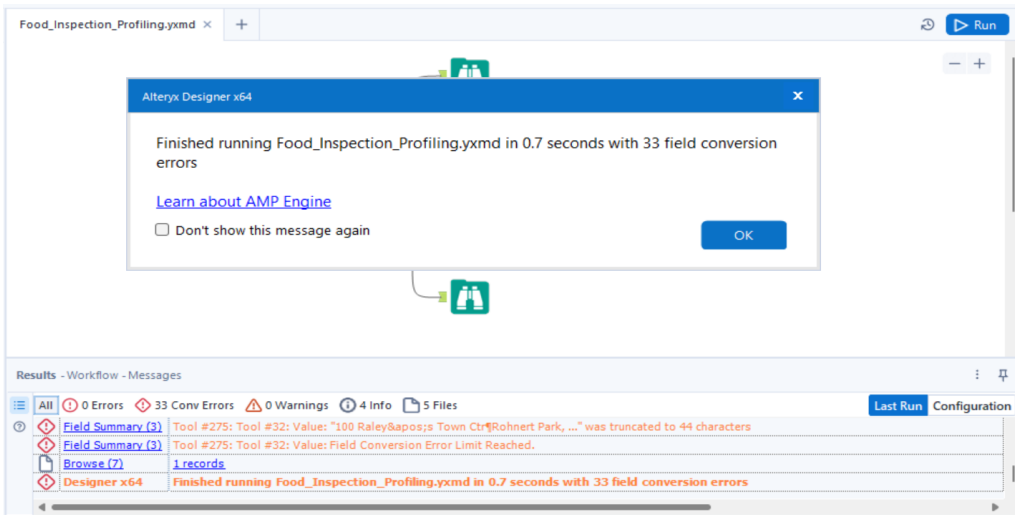
Team Member Group 13:

1. Shreyasi Wakankar : 002771284
2. Muskan Srivastava: 002794929
3. Pramita Dileep Sandhyan: 002766881
4. Rhea Bajpai: 002702927

Alteryx Profiling:



Time taken for execution:



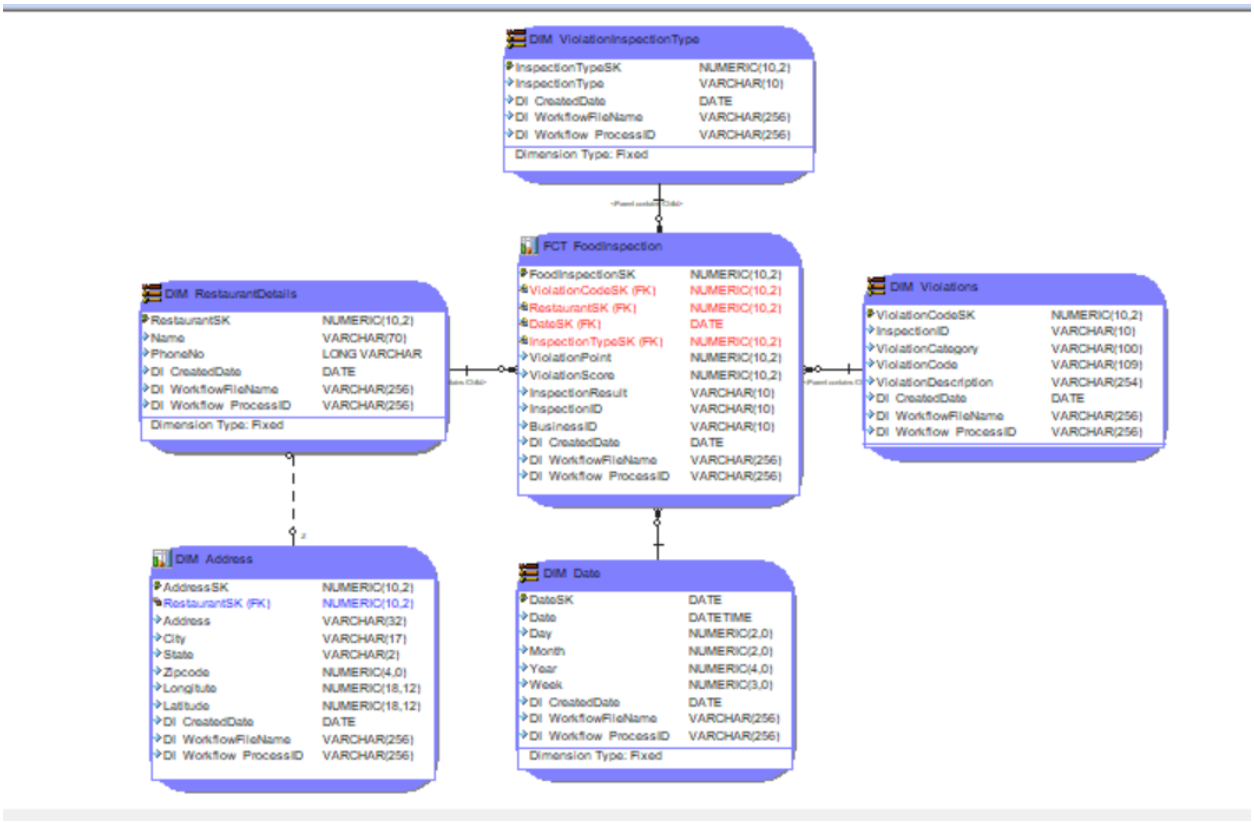
Observations and Changes Required:

Field Name	Data Type from website	Data Type after importing csv	Data Type Changes Required after profiling	Longest Length	Null Values
Business Id	String	String	String	9	0
Name	String	String	String	70	0
Address	String	String	String	32	0
City	String	String	String	17	0
State	String	String	String	2	0
ZipCode	String	String	Int	10	0
Phone No	String	String	Long	12	4307
InspectionID	String	String	String	9	0
Date	Date & Time	String	Date & Time	10	3
Inspection Type	String	String	String	9	3
Violation Code	String	String	String	109	10082
Violation Description	String	String	String	254	10132
Location	Location	String	String	86	3

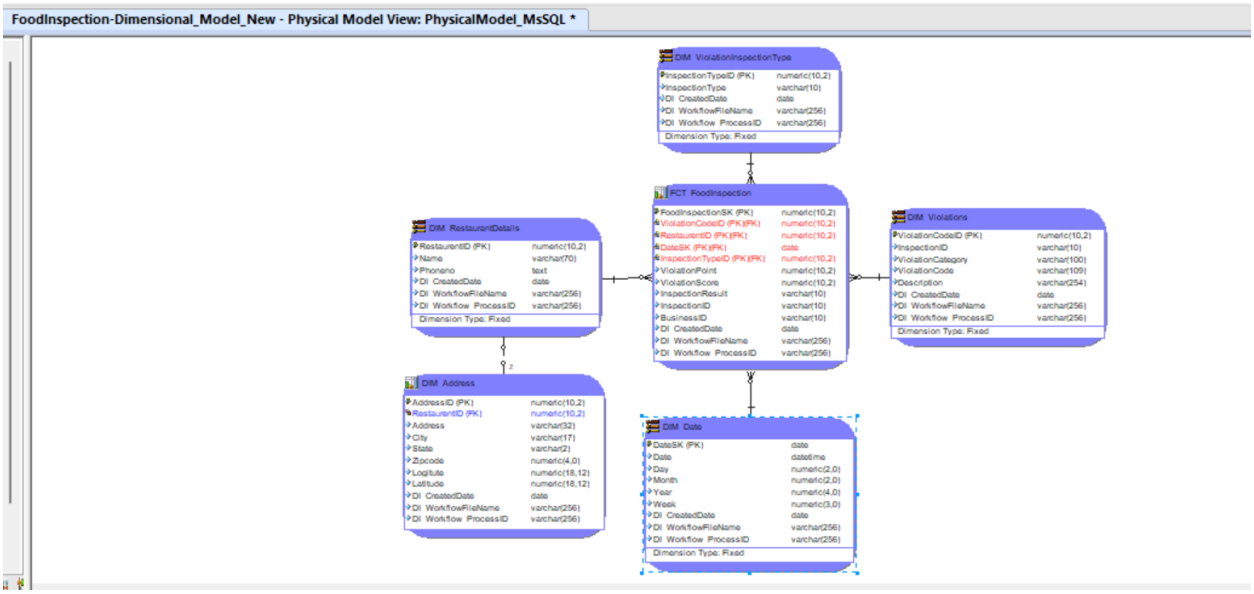
We observe that there should be three changes in data type for three columns: ZipCode, PhoneNumber and Date. There are three columns (PhoneNumber, ViolationCode and Violation Description) which have null values.

Part 2- ER Studio

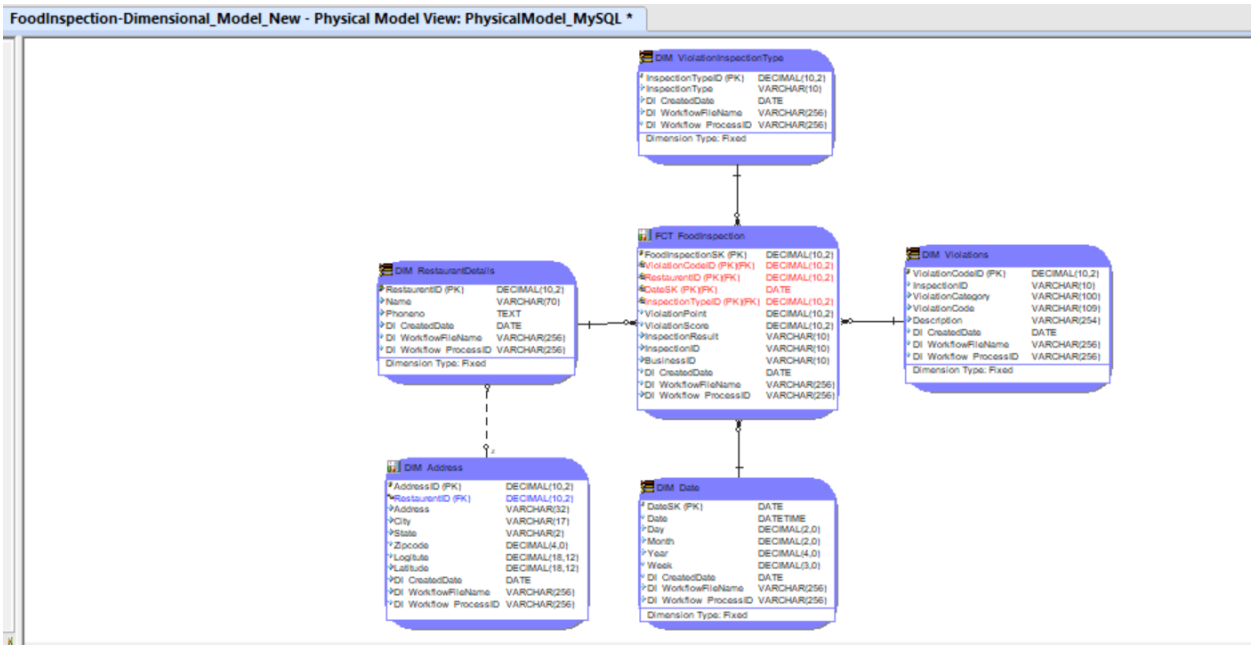
Logical Model:



MSSQL Physical Model

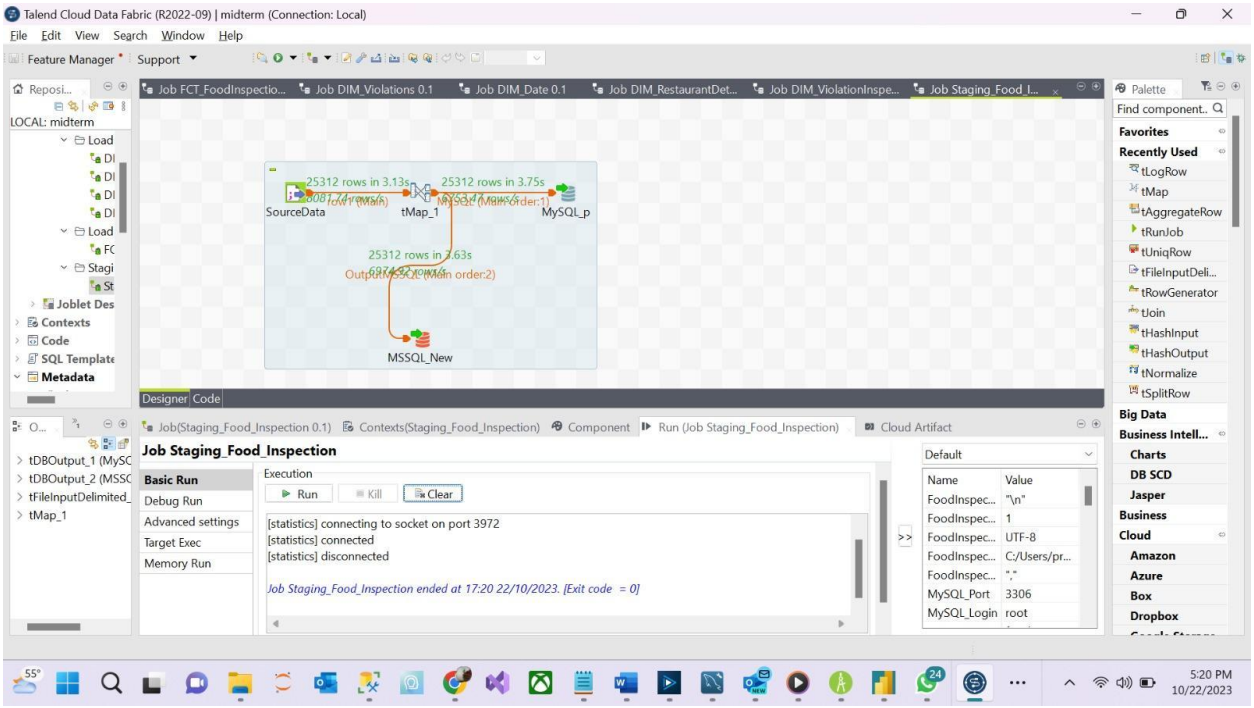


MYSQL Physical Model



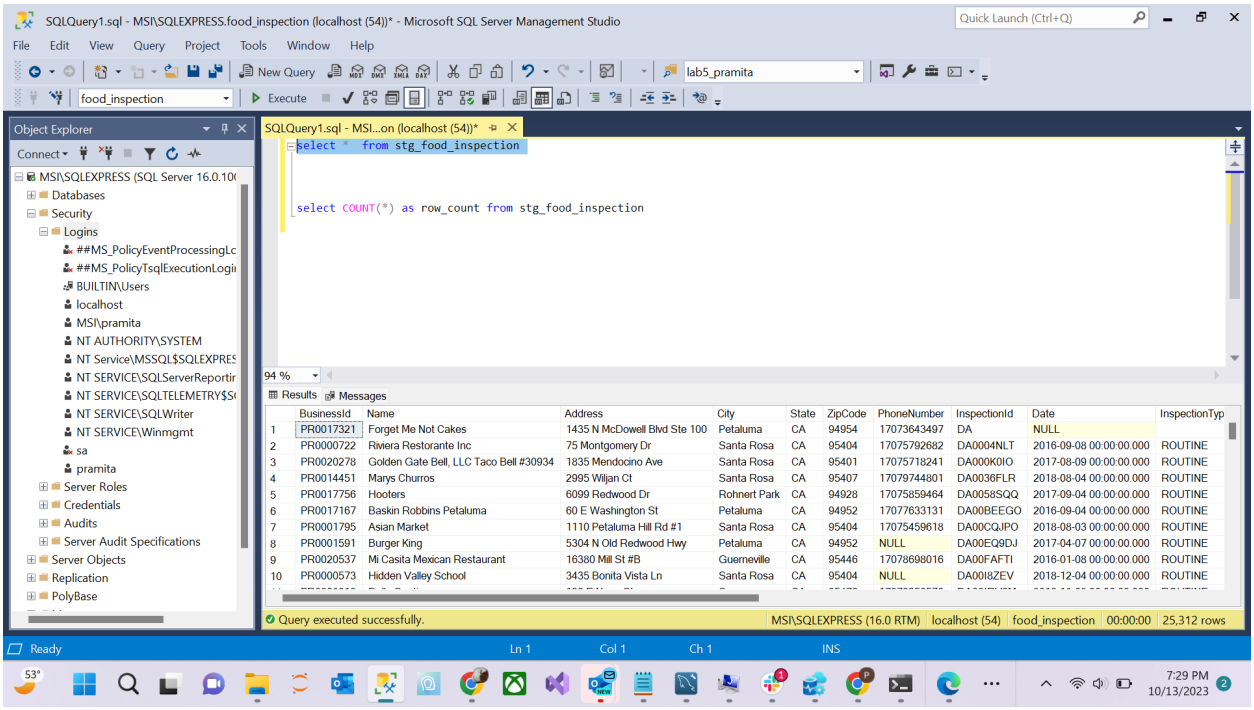
TALEND:

Creation of Staging table:



Purpose and Business Logic: The source data is cleaned and fed into MySQL and MSSQL through a Tmap so that we can join rows, alter the rows and perform transformations. The data has been taken from the source provided in the class and it has been profiled in Alterx as shown above.

Table in MsSQL:



Row count:

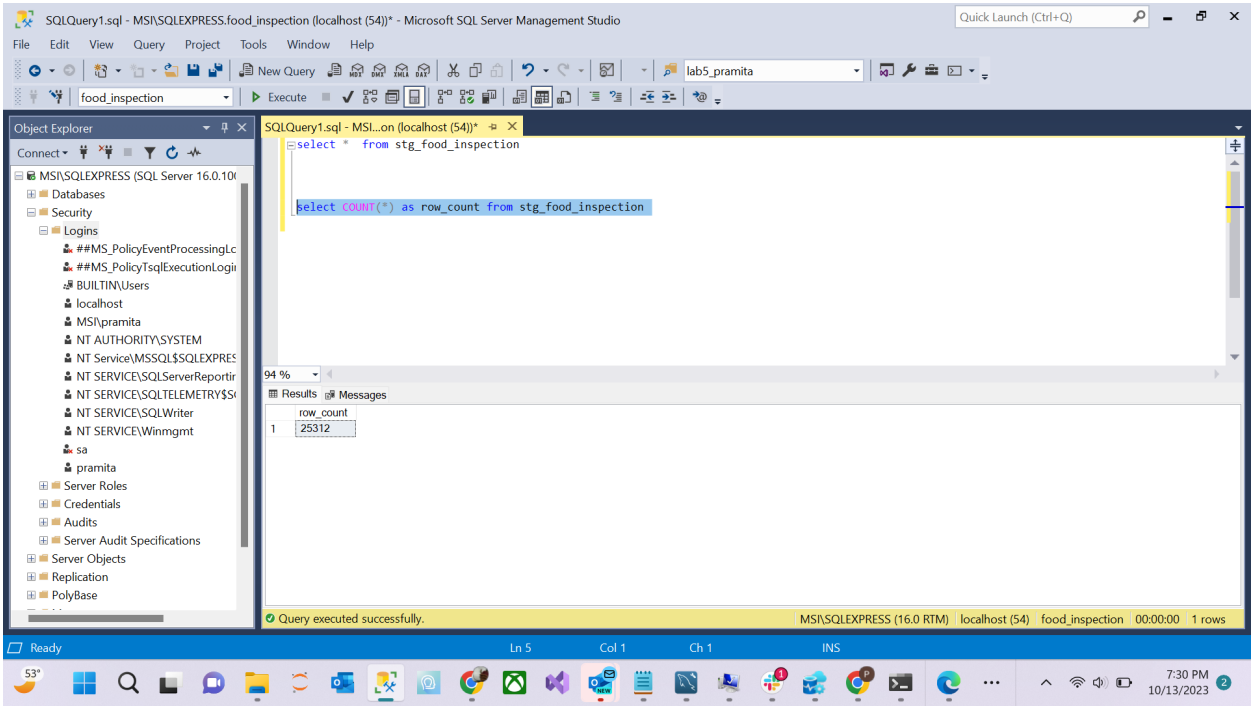
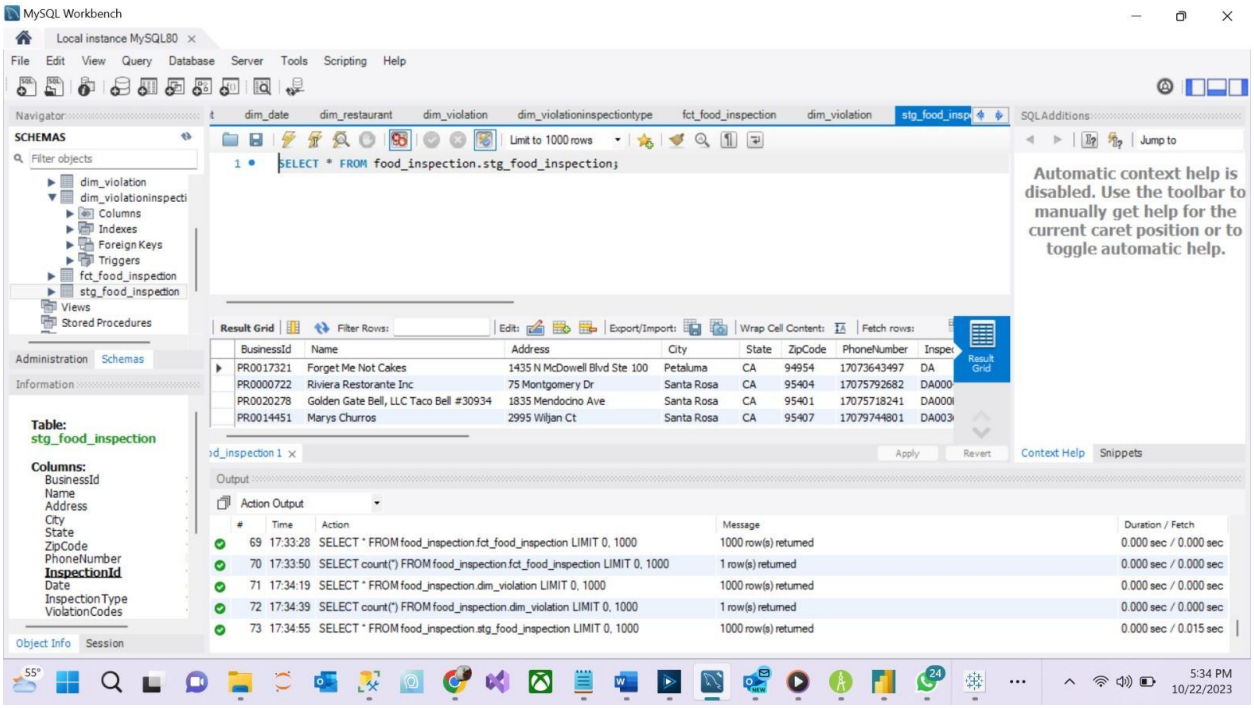
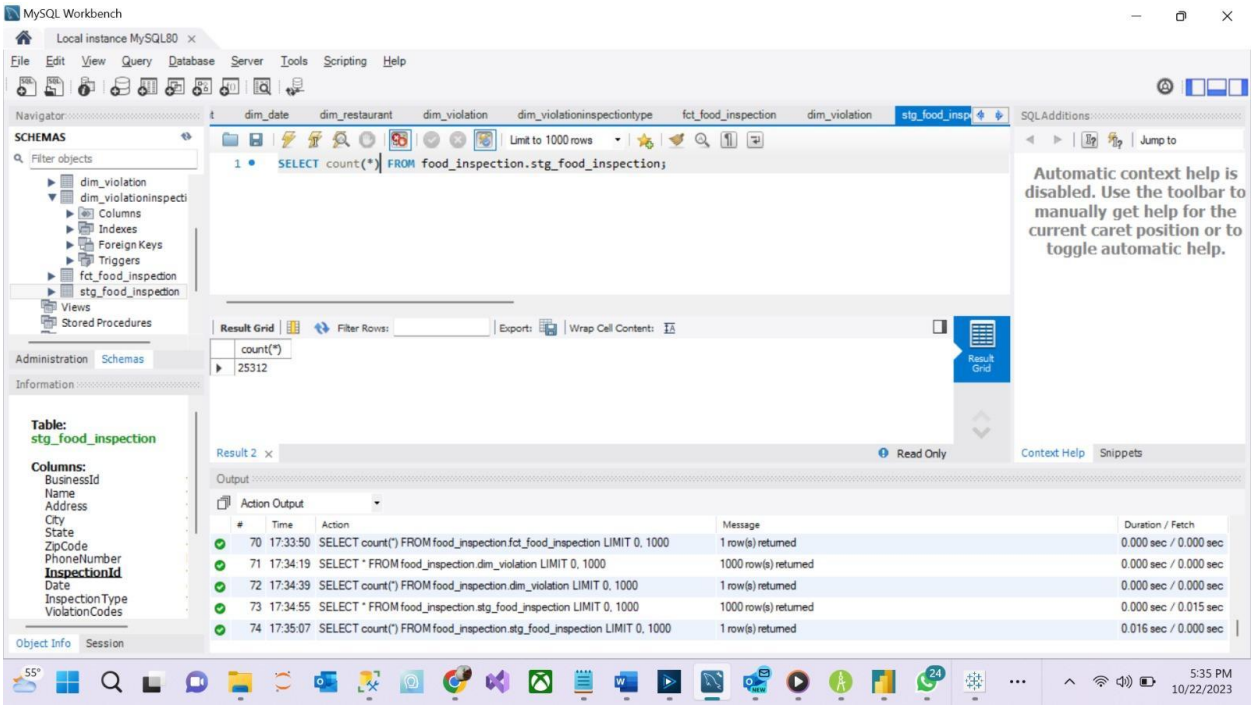


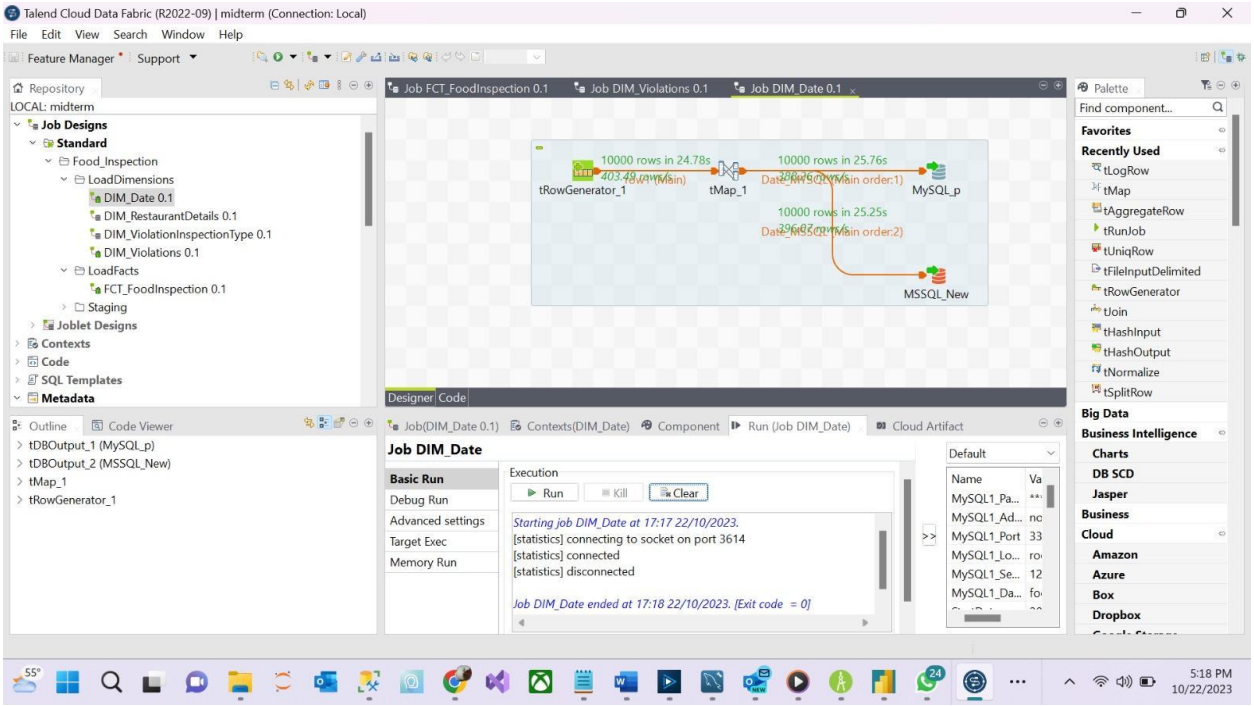
Table in MySQL:



Row Count:



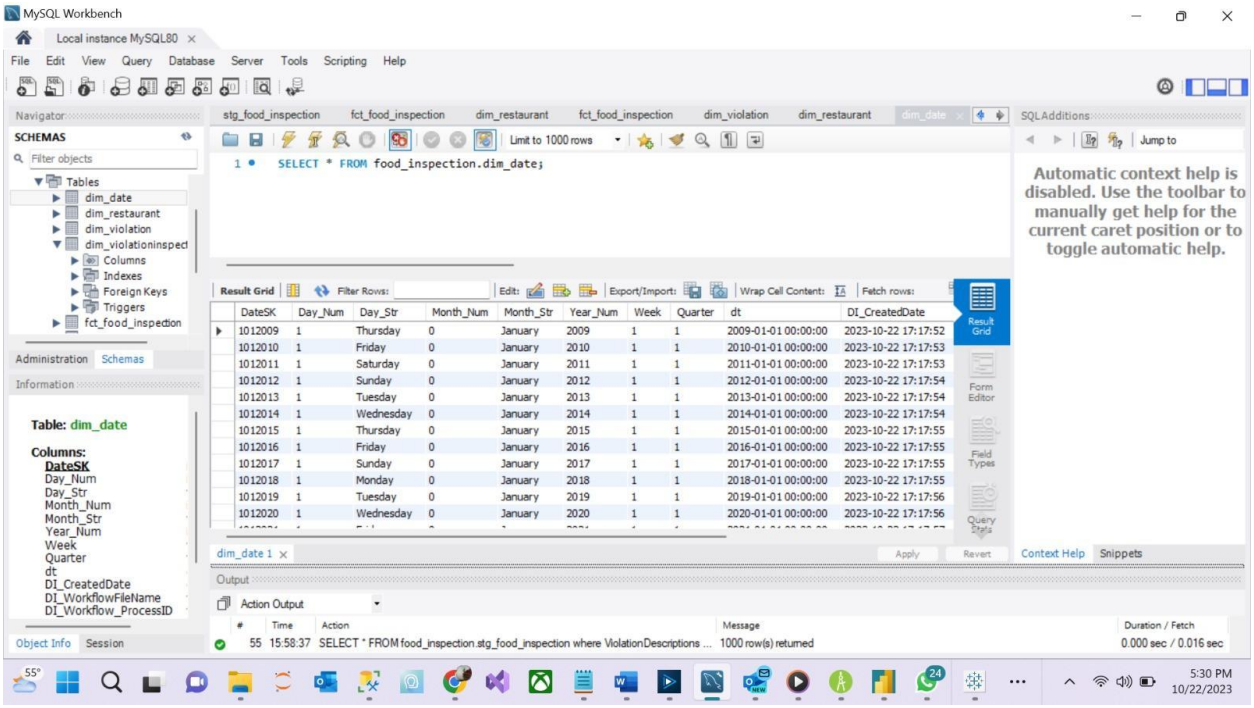
Loading Date Dimension:



Purpose and Business Logic: We have used a tRowGenerator to load the data to MYSQL and MSSQL through a tmap. We have used a tRowGenerator so that we can generate rows and field as needed and feed each filed with a value taken randomly from the provided list. Tmap is used to put the fields and rows

generated from tRowGenerator to both the databases by manipulating/altering the input. We have given the grain of the date_dimension as Day, Month, Year, week, Quarter.

Table in MySQL:



Row Count:

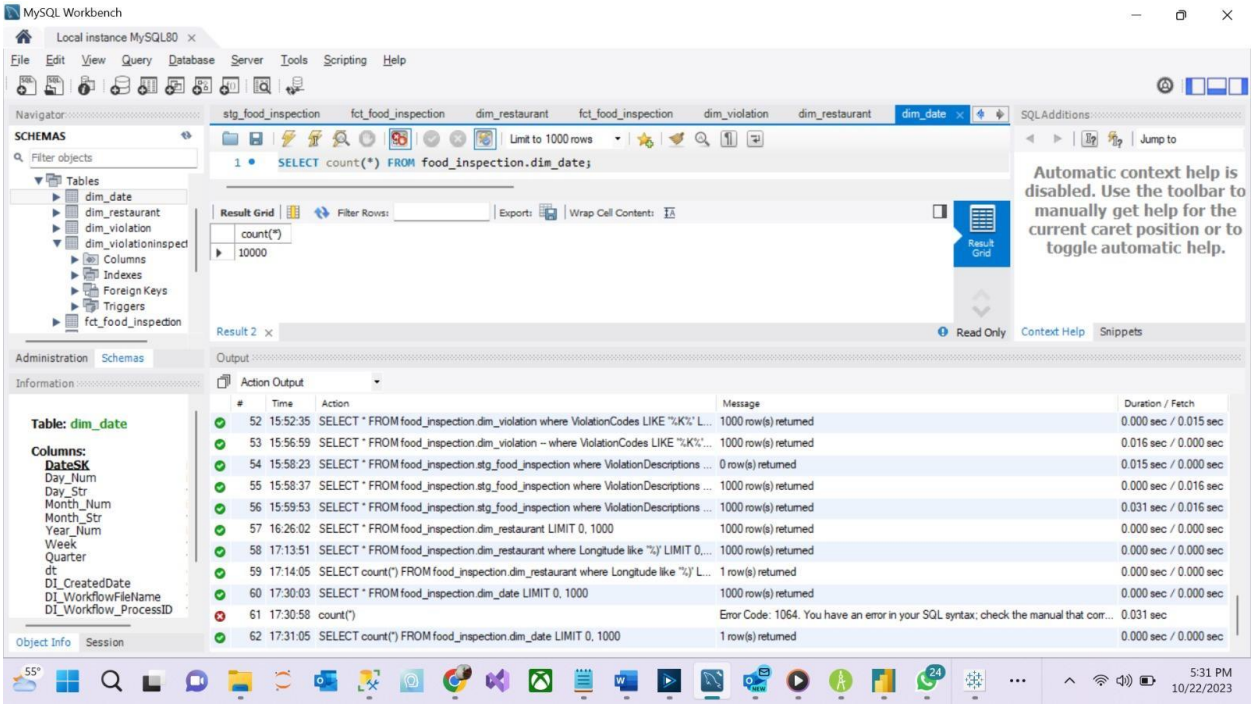
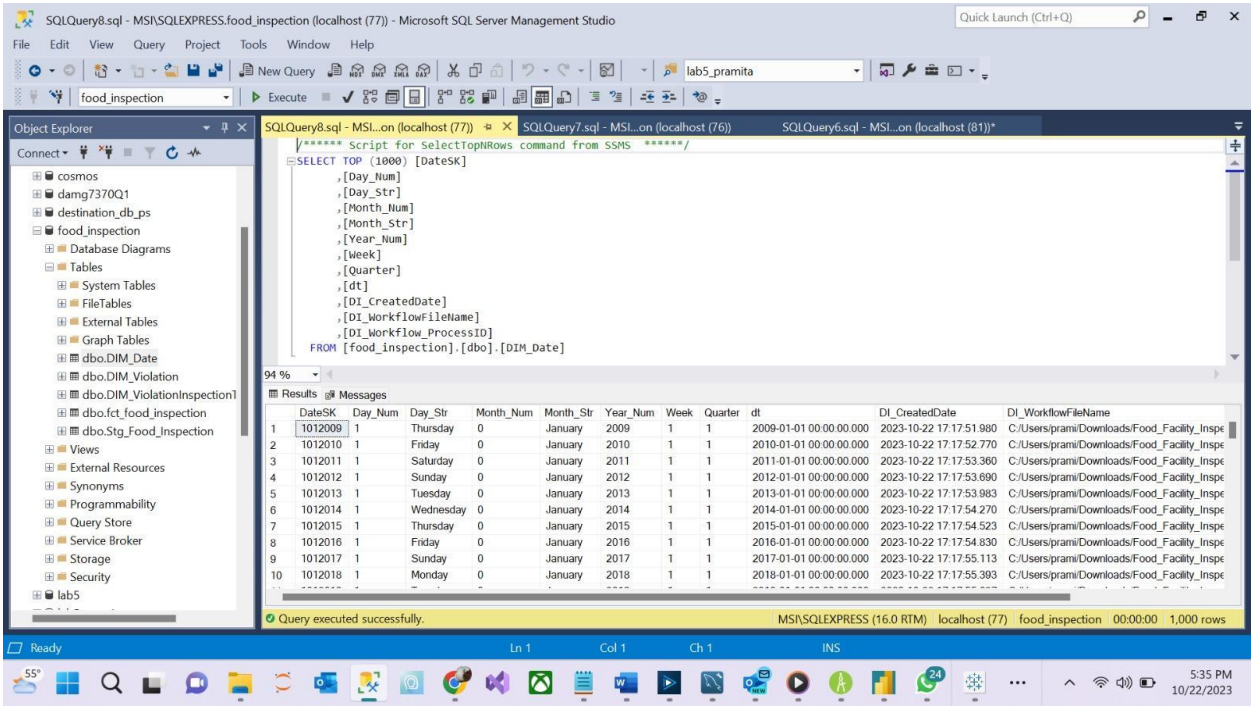
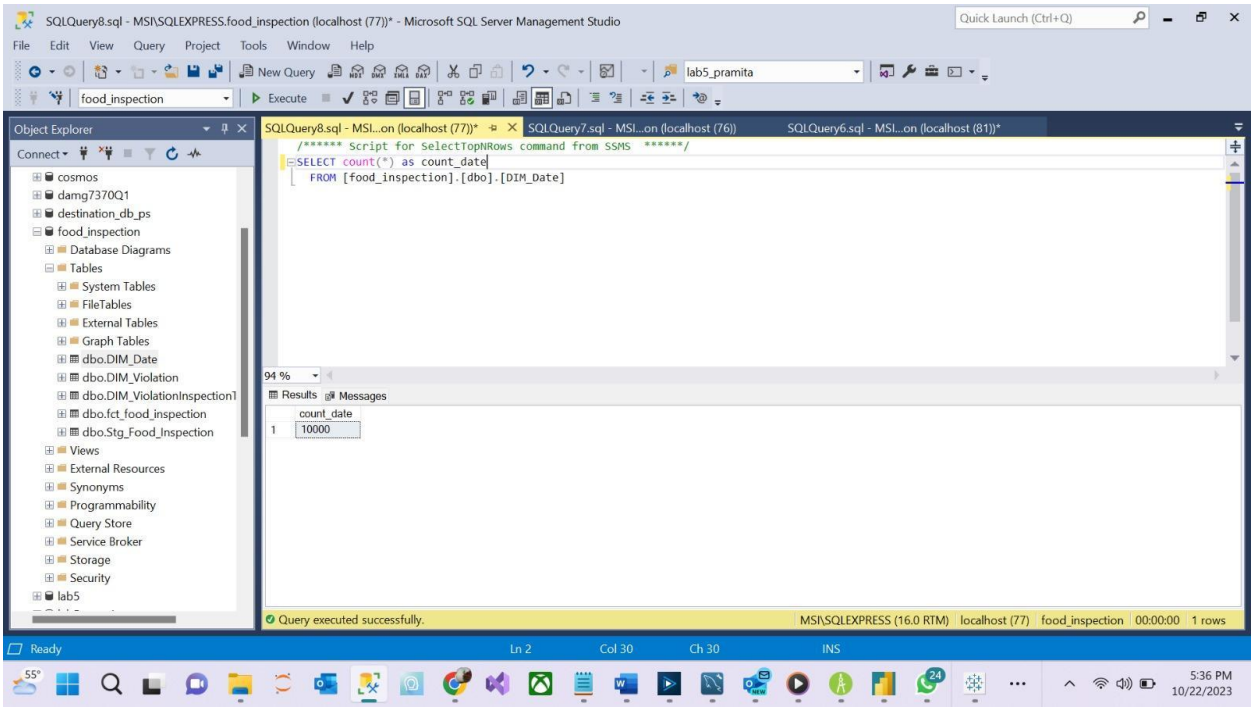


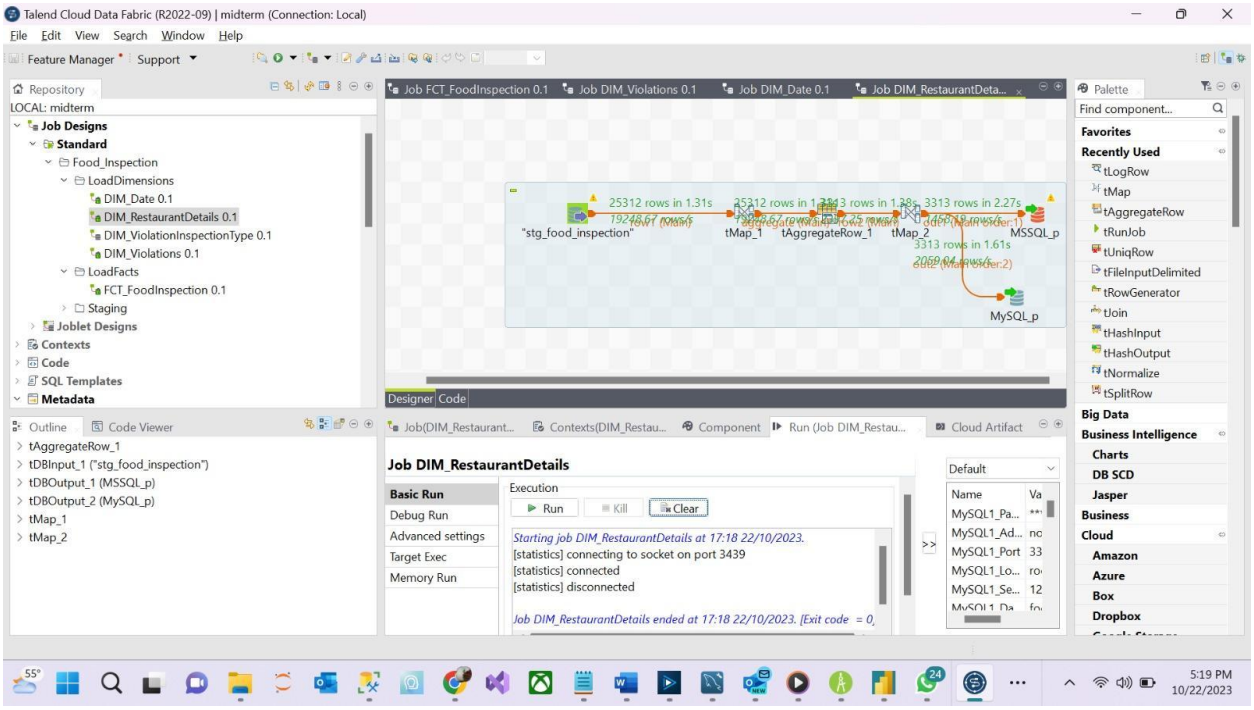
Table in MsSQL:



Row Count:

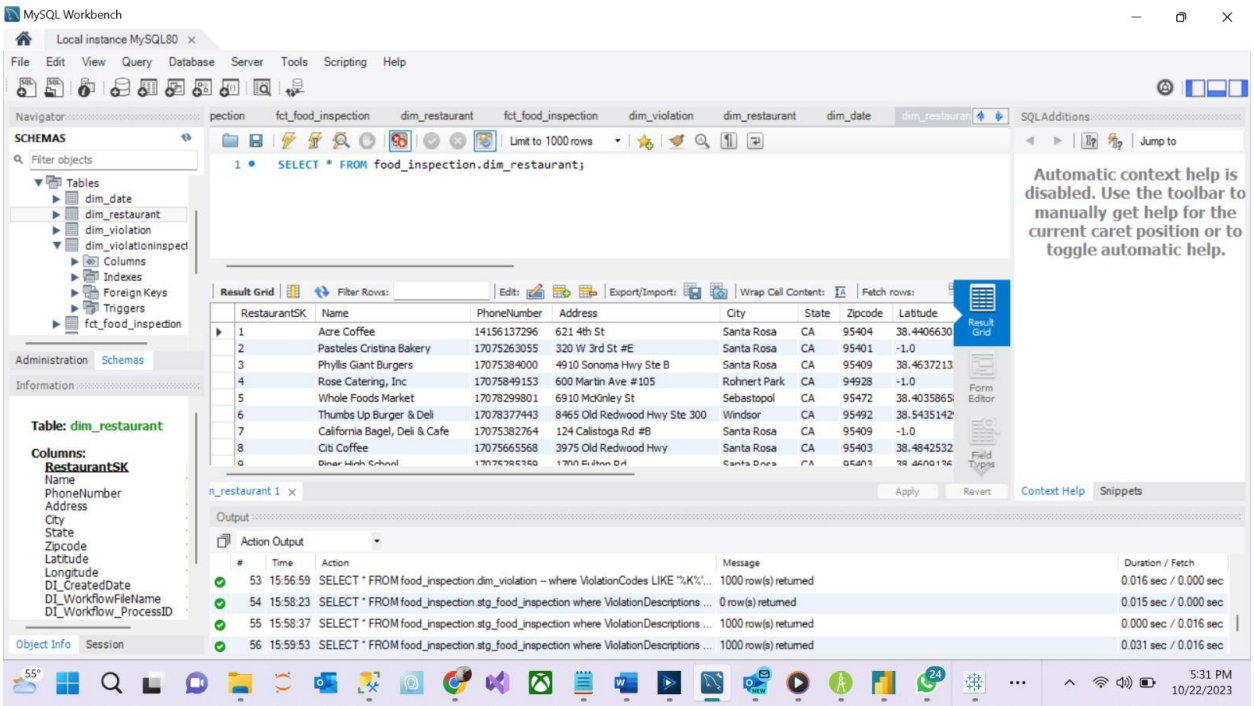


Loading Restaurant Dimension:



Purpose and Business Logic: The staging table is taken as an input and fed into the Tmap. The output from Tmap is taken as an input to TaggregatorRow so that we can perform function similar to SQL groupby. It receives the flow and aggregates it based on Name, Phone no, Address, City, State, Zipcode, Latitude and Longitude. We have taken the Latitude and Longitude as separate rows to have a more well defined grain. The output is fed into another trow fed into the dbs.

Table in MySQL:



Row count:

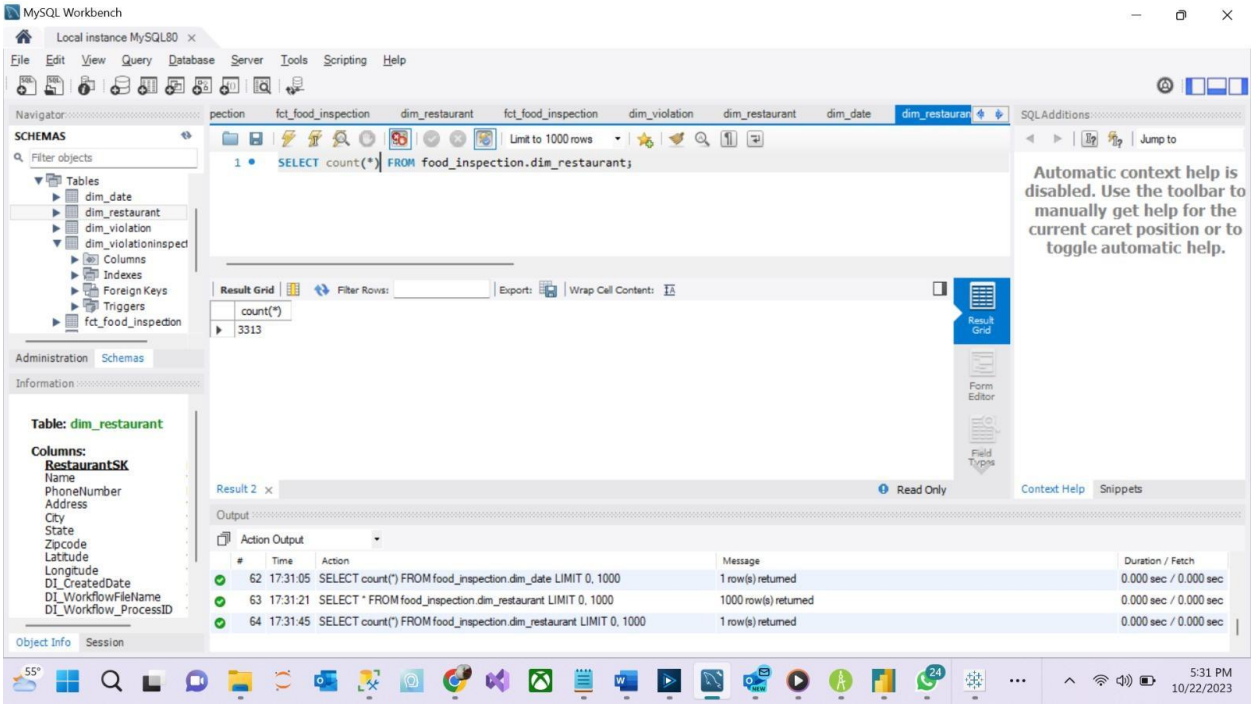
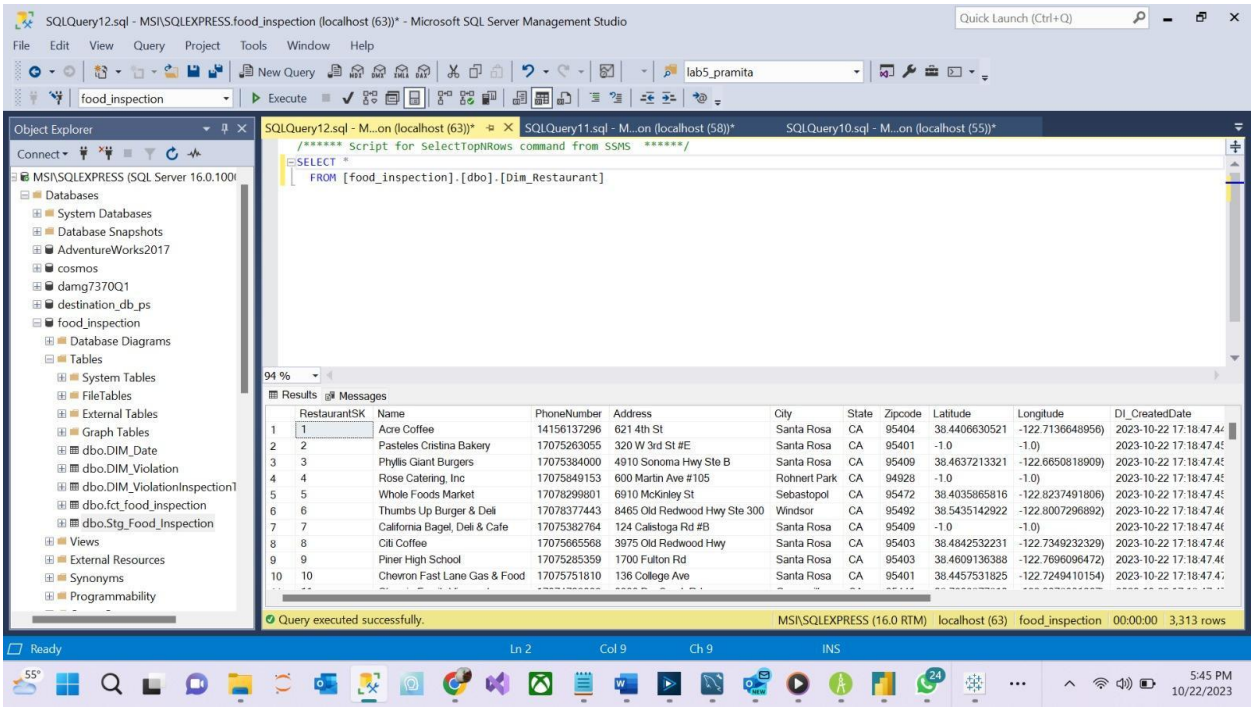
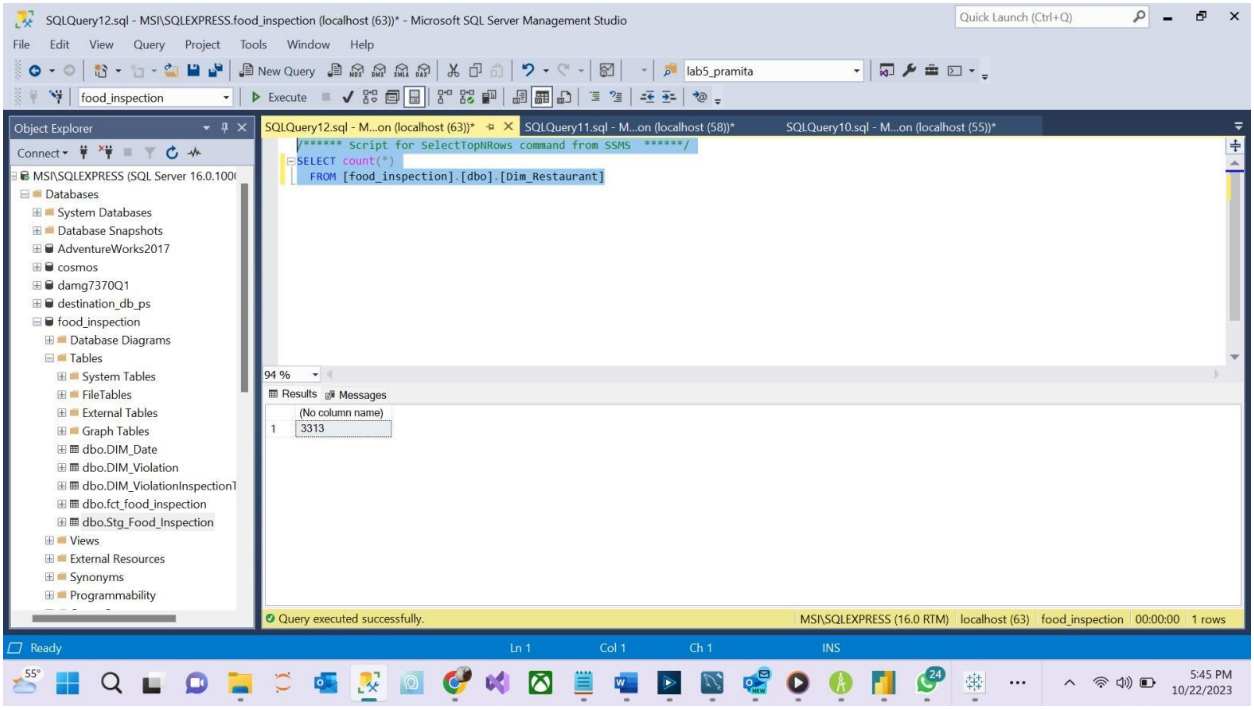


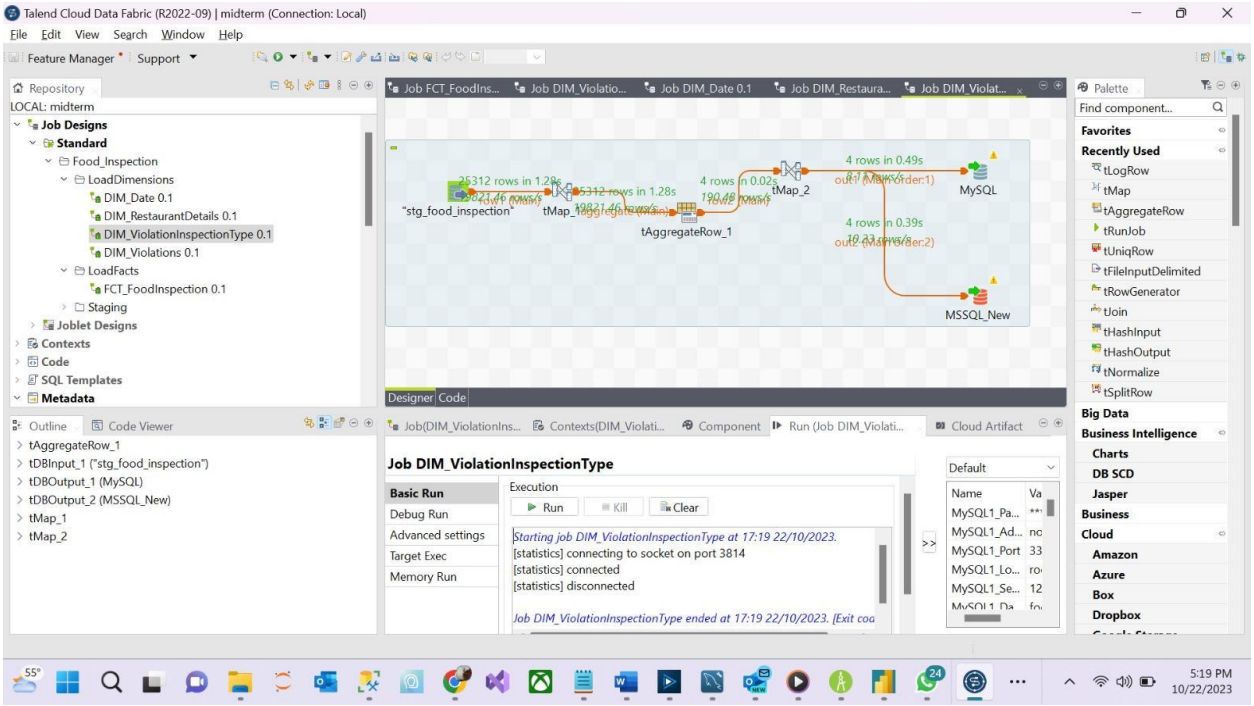
Table in MsSQL:



Row Count:



Loading Violation Inspection Type Dimension:



Purpose and Business Logic: The input is taken from stg table and fed into Tmap and Taggregaterow subsequently. We have fed Inspection type in the Taggregaterow to be taken as the field to be grouped. It is being fed as an input to tmap and further to the DBs. The output we receive from this dimension is the types of inspection which are: Routine, Followup, Complaint and Null.

We have done this so that we will get the information about the types of inspection and further used in visualization as shown below.

Table in MySQL:

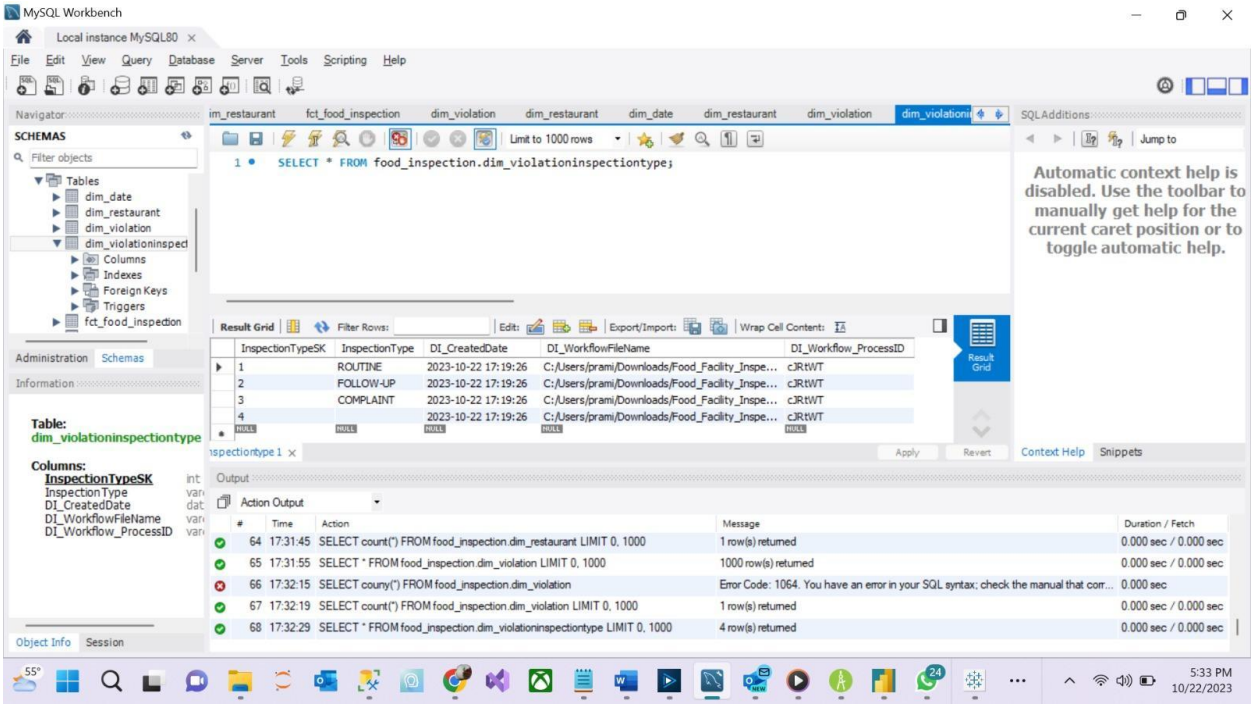
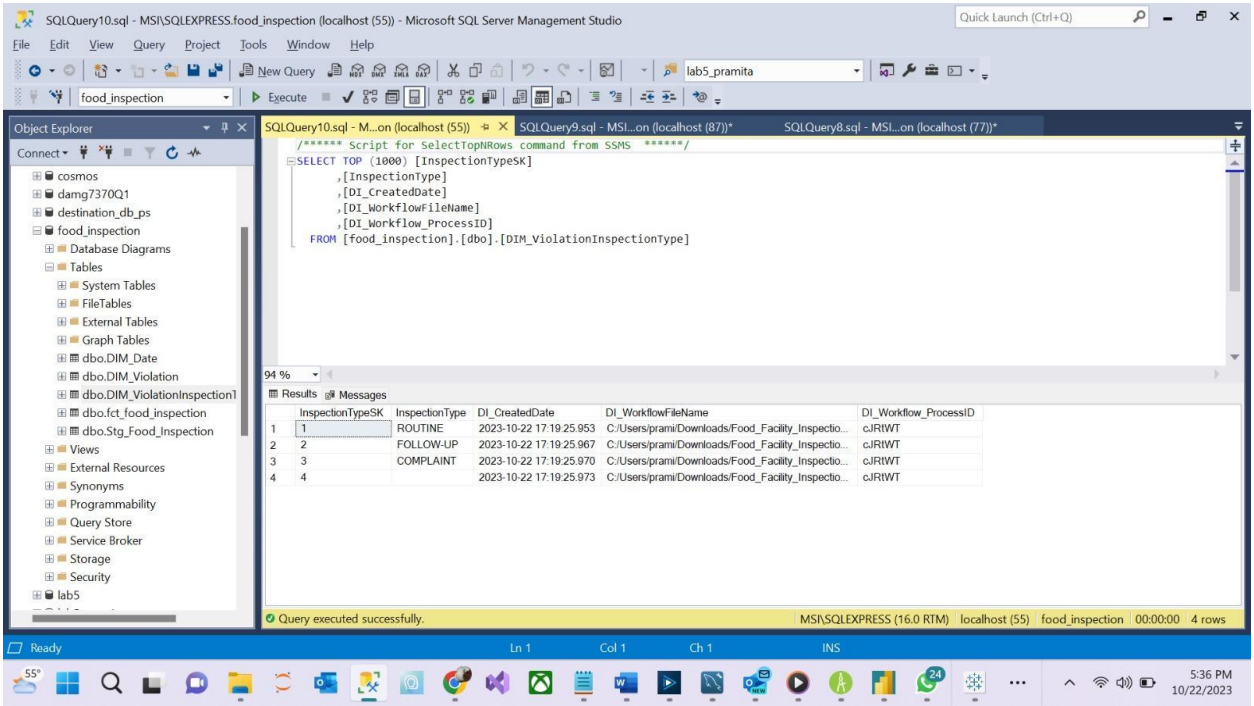
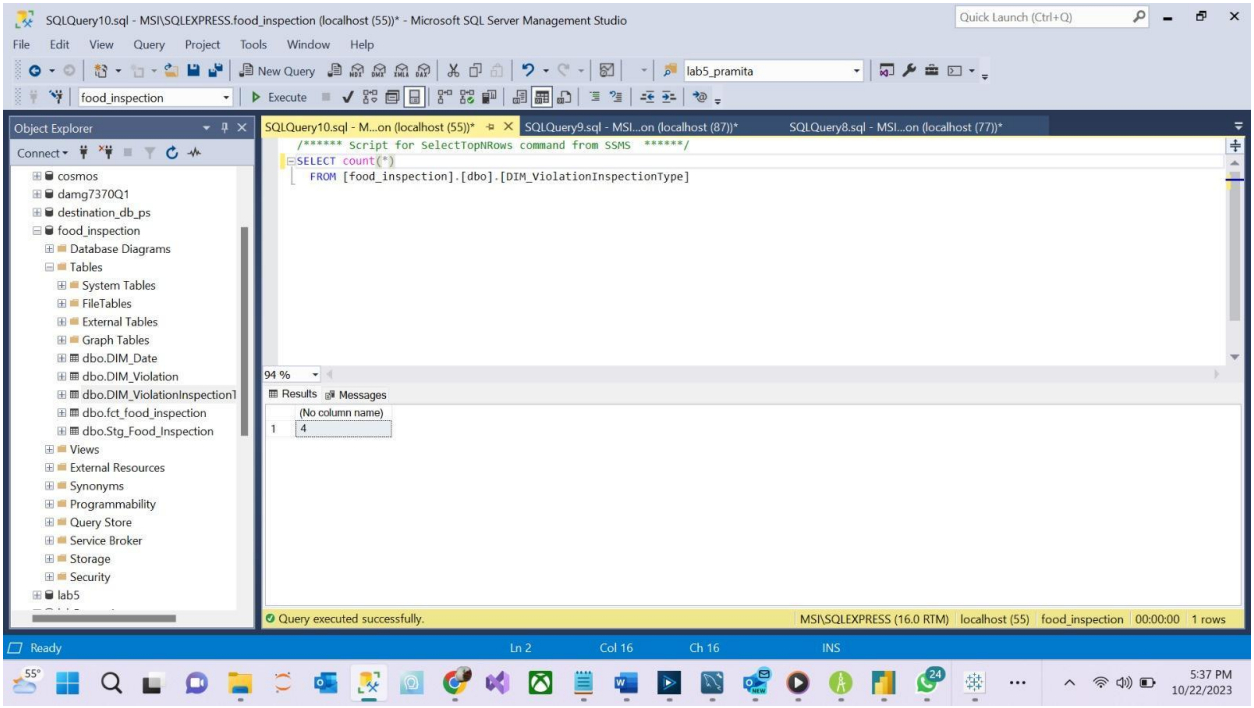


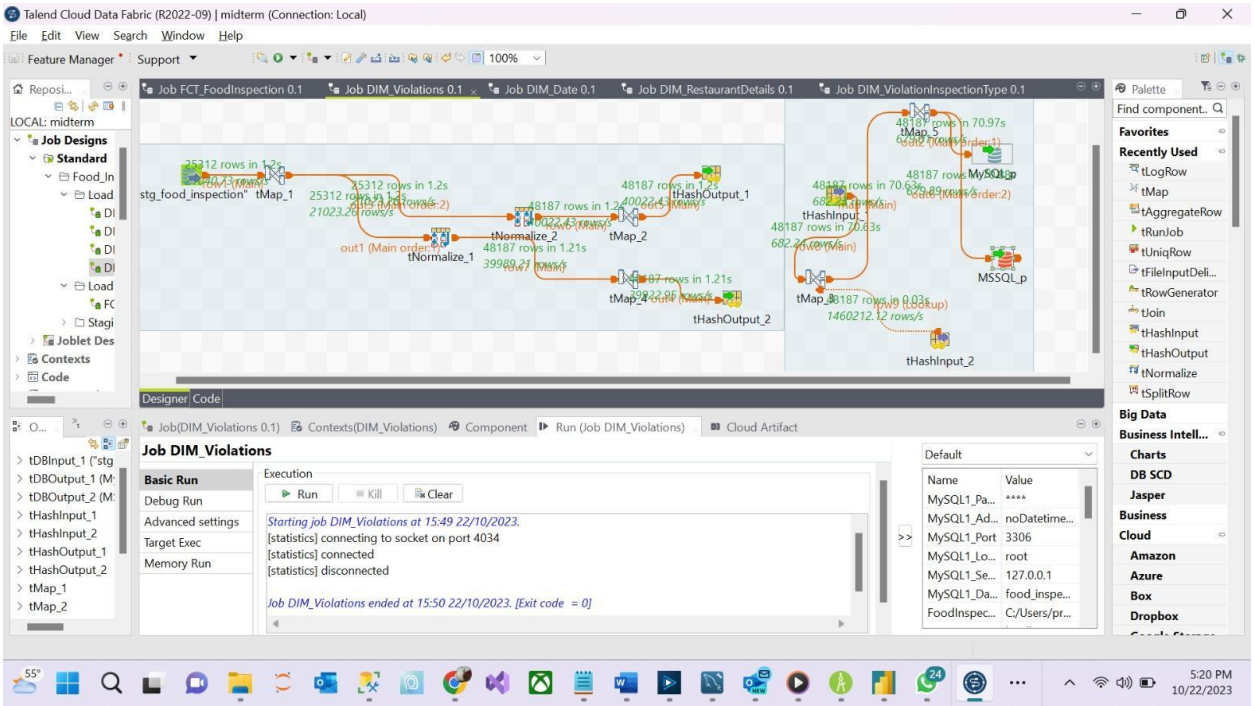
Table in MsSQL:



Row Count:



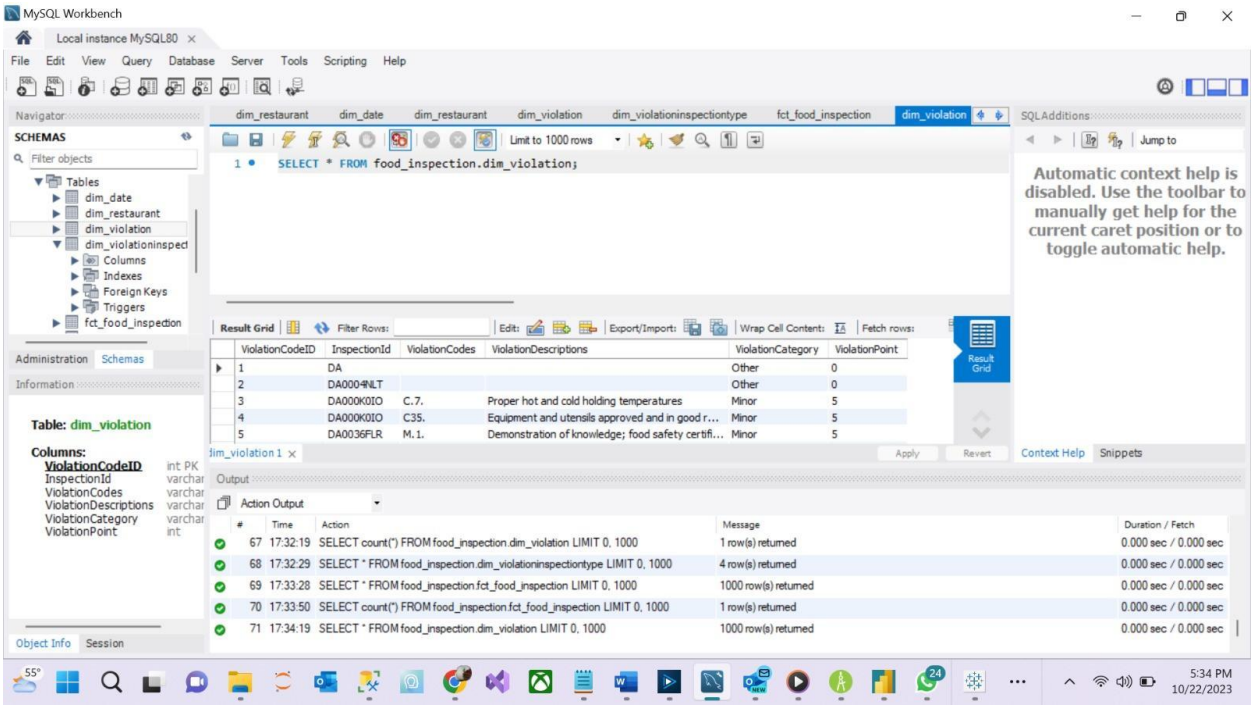
Loading Violation Dimension:



Purpose and Business Logic: We have used 2 Tnormalize components to normalize 2 multivalues columns: ViolationCode and ViolationDescription. On normalizing both the columns, we have passed the normalized values to Hashoutput 1 and Hashoutput2. Further, we are feeding the outputs to Hashinputs. We have then used Tmap to merge the data into the single table and store it in the DB. We have done this to normalize the

multivalued columns and store the data into one single table and were required to get the ViolatioCategory and ViolationScore from the merged values.

Table in MySQL:



Row Count:

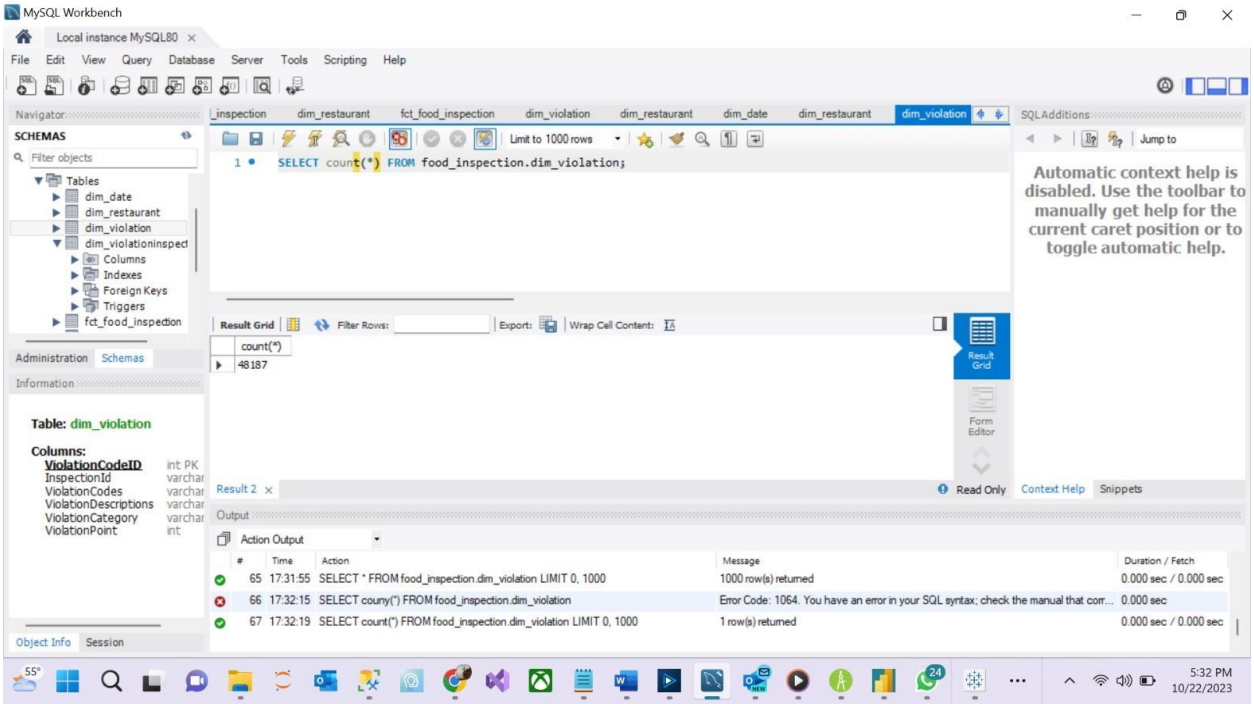
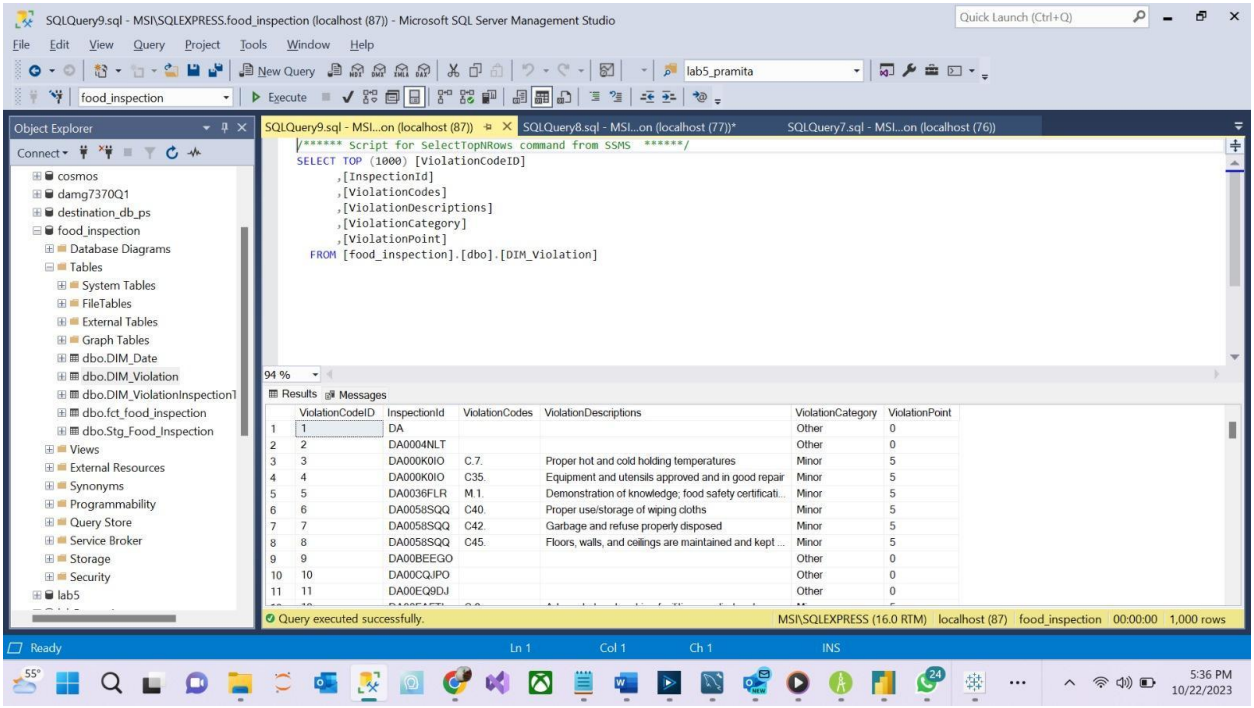
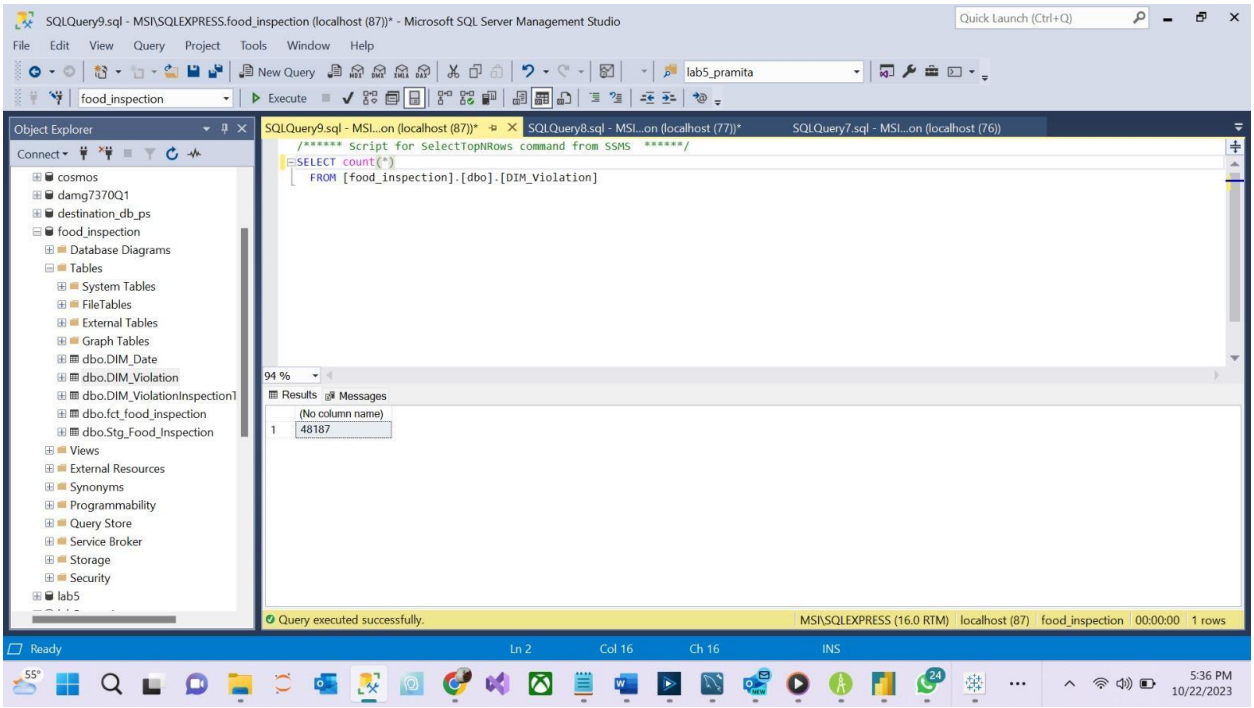


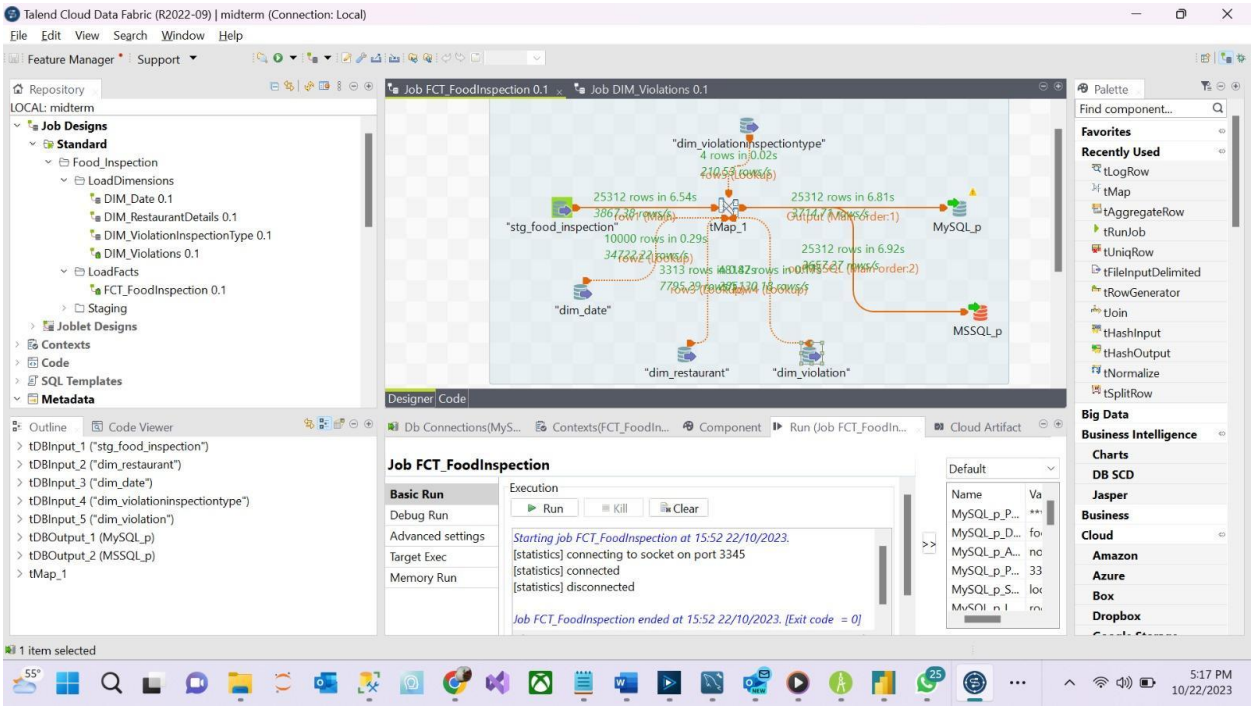
Table in MsSQL:



Row Count:

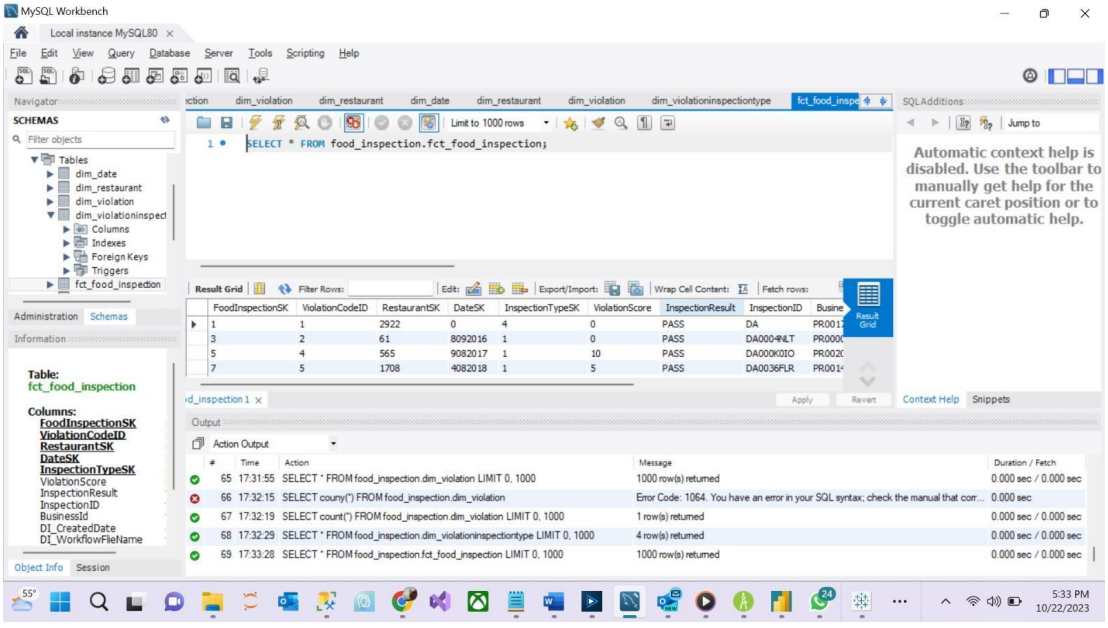


Loading Fact table:



Purpose and Business Logic: The purpose of the fact table is to store all the Primary Keys together in a table as a Surrogate key and organize all the quantitative data in order to provide a solid foundation for decision-making in a business intelligence. We have used the inputs from staging table, and all the dimensions such as Date dimension, Violationinspectiontype dimension, violation dimension, restaurant inspection. We have fed all of them to tmap which is fed to the DBs. As the final output in the DB, we get FoodInspection SK, ViolationCodeID, Restaurant SK, Date SK, InspectionType SK, Violation score which is the addition of all the violation category, Inspection result which is pass or fail based on If violation score is between 0 and 60 is PASS and violation score > 60 is FAIL, InspectionID and BusinessID.

Table in MySQL:



Row Count:

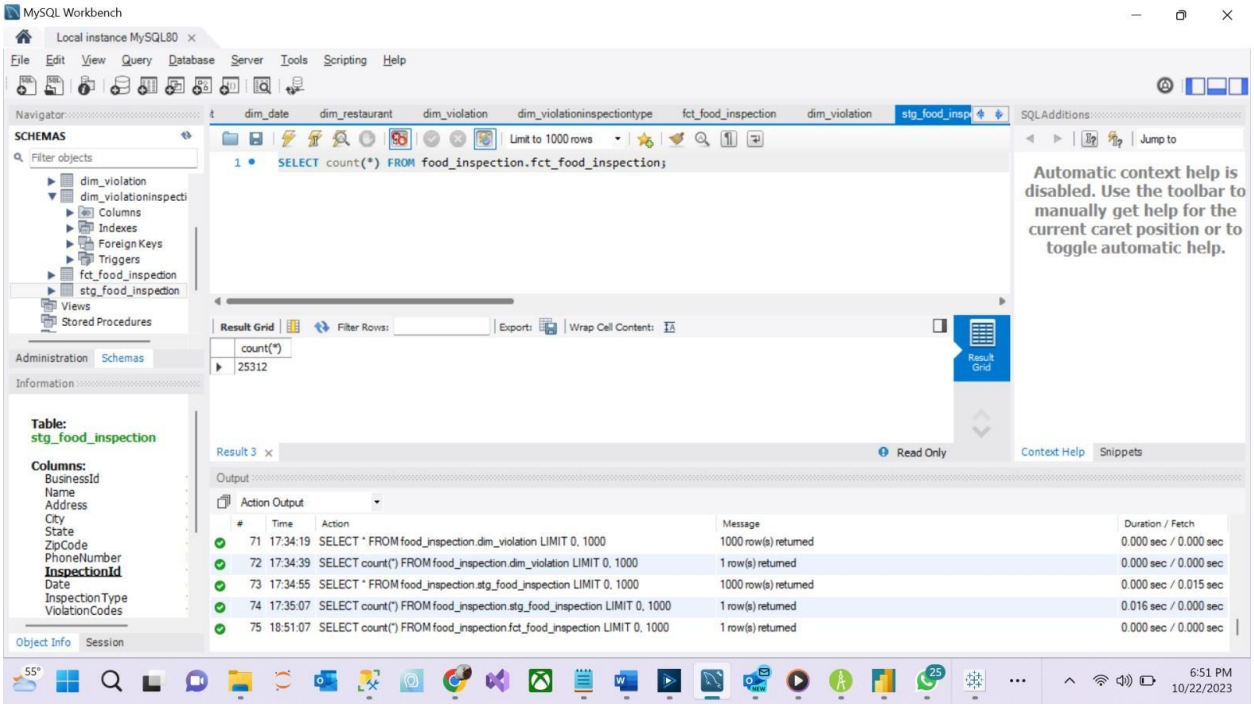
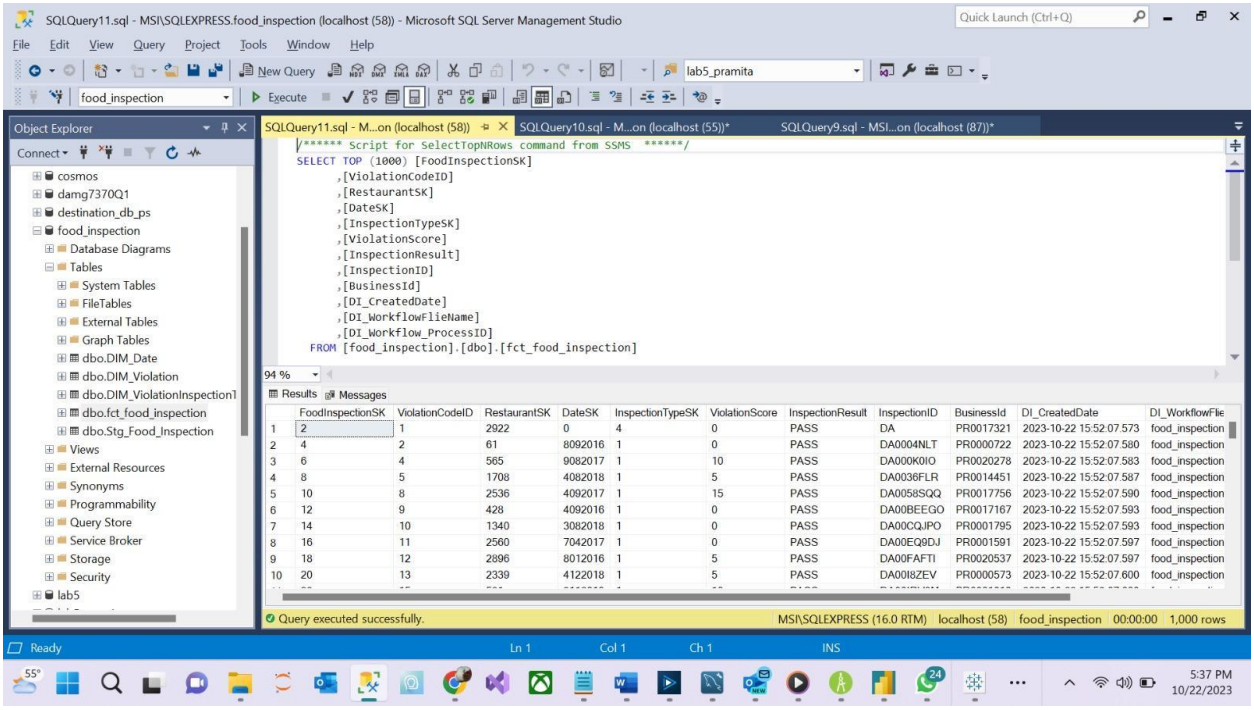
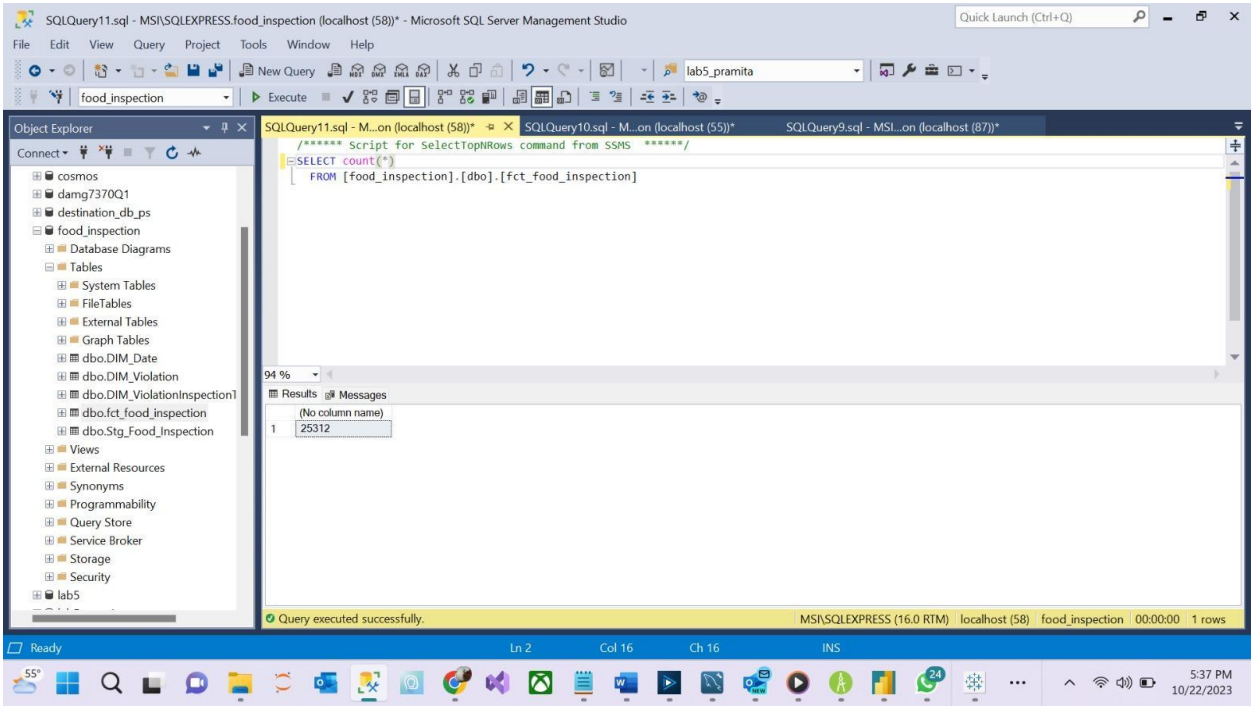


Table in MsSQL:



Row count:


SQLQuery11.sql - MSI\SQLEXPRESS.food_inspection (localhost (58))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
lab5_pramita
food_inspection
Execute
Object Explorer
Connect
cosmos
damg7370Q1
destination_db_ps
food_inspection
Database Diagrams
Tables
System Tables
FileTables
External Tables
Graph Tables
dbo.DIM_Date
dbo.DIM_Violation
dbo.DIM_ViolationInspection1
dbo.fct_food_inspection
dbo.Stg_Food_Inspection
Views
External Resources
Synonyms
Programmability
Query Store
Service Broker
Storage
Security
lab5
SQLQuery11.sql - M...on (localhost (58))*
SQLQuery10.sql - M...on (localhost (55))*
SQLQuery9.sql - MSI...on (localhost (87))*
/****** Script for SelectTopNRows command from SSMS ******/
SELECT count(*)
FROM [food_inspection].[dbo].[fct_food_inspection]
94 %
Results
Messages
(No column name)
1
25312
Query executed successfully.
MSI\SQLEXPRESS (16.0 RTM)
localhost (58)
food_inspection
00:00:00
1 rows
Ready
Ln 2
Col 16
Ch 16
INS
5:37 PM
10/22/2023