**NYC TAXI TRIPS**

This dataset was chosen because of the well-defined data dictionary and the richness of the data across multiple years. The total size of the dataset was too huge to be analyzed by Excel. Therefore, the first step was to create a SQL database, create a table and insert the data. This provided a perfect opportunity to showcase SQL and PowerBI skills. It’s going to be challenging but fun.

Steps.

1. Study the data dictionary to understand the data types.

* The data types were identified based on the data dictionary definitions and observations from the data tables to place best match.
* Assumption: The description from the data dictionary table was correctly defined.

1. Create SQL Server database

* Create Database named TRIPS

1. Create a table NYC\_Trips and Taxi\_Zones

* Initially, one table was created for all the years. It was later discovered that the data for each year had extra/ less columns.
* Solution: Create a table for each year based on the data columns for each data file per year.

1. Import the data

* The first step was to convert all fields to VARCHAR
* Second, import the data using bulk import query

1. Explore the data in SQL Server database
2. Aggregate the data before importing it into PowerBI
3. Create data models

Challenges & Solutions

|  |  |  |
| --- | --- | --- |
| Serial No | Challenge | Solution |
| 1. | On Insert into Taxi\_Zones table, fields ‘Borough’, Zone\_Name’ & ‘Service\_Zone’ had limited characters as it was created as a VARCHAR only | Created an ALTER Query to increase the size of characters to 500 |
| 2. | Altering multiple columns gave an error | Created separate ALTER queries for each field. |
| 3. | Large data files to GitHub. | The data files were too large to be uploaded to GitHub, the workaround was to upload to OneDrive on this link: [NYC\_TRIPS](https://afzdxb-my.sharepoint.com/:f:/g/personal/innocent_kyalo_alfazance_com/EiNppSKimd9JiAdfNknRNN0B3dGRIW0c0ZiDgrwPUzzSIA?e=xFGI0y) |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Challenges & Solutions**

1. Import of Data

* The raw data for the Trips table was voluminous to be inserted into the table using.
* Workaround 1
  + At first, I tried an BULK insert query which kept on giving errors about a mismatch between my table data type and my CSV data types. The initial BULK insert is in the SQL Queries folder named ‘Bulk Insert Trips Data Query’
* Workaround 2
  + After no success with workaround 1, I researched a technique that proved to work. It involved altering my existing fields data type to VARCHAR using query ‘ALTER TRIPS TABLE \_ ALL FIELDS TO VARCHAR’, importing the data and finally converting them back into the original data types.

1. Assuming all Trips Data has the same data fields.

* Initially it was assumed that all data files have the same number of data fields, and it was ordered in the same format. Upon investigation, I discovered that this was not consistent for all the years.
* Next Step: Create separate tables for each year and import the data separately to avoid inconsistencies.