We began our project by obtaining the data dictionary from the 311 website, as provided by the Professor. Additionally, we accessed JSON files to gather information on living wages in each NYC county. To ensure easy data transfer to Microsoft Azure, we utilized Python scripts to convert the JSON files into CSV format. We focused on three specific occupations - IT, Management, and Healthcare - filtering their annual salaries. Utilizing the data dictionary from the 311 site, we constructed a dimensional model with dimensional tables, aligning them with the columns in the data dictionary. These tables contained attributes corresponding to the column names in the data dictionary.

Once the dimensional model was completed, we proceeded to set up accounts on both Azure cloud and Snowflake. Azure served as the repository for our CSV files and data, stored in blobs, which could be seamlessly integrated with Snowflake through scripts concerning container names and contents. Snowflake served as the platform for hosting our dimensional model and facilitated integration with PowerBI for data visualization. However, we encountered challenges while attempting to upload CSV files to Azure and establish the necessary link to Snowflake. Due to time constraints, we had to devise an alternative approach to import the data into PowerBI for visualization. Consequently, we merged the 311 data frame with the living data frame and extracted the columns corresponding to the dimensions in our dimensional table. We imported the CSV file into PowerBI to generate visuals displaying the annual salary lists for the three occupations in each borough.

To access the CSV file, we integrated the 311 website's API into our Python script, enabling us to retrieve the data and create a data frame. This data frame encompassed fields such as created date, closed date, agency, agency name, address, zip code, borough, city, complaint type, complaint description, and unique key. We merged this data frame with the one derived from the 311 CSV file. Additionally, we utilized Python scripts to leverage the Zillow API and acquire up-to-date information from the Zillow website. The data obtained consisted of address, boroughs, zip code, number of bedrooms, number of bathrooms, and price ranges.



