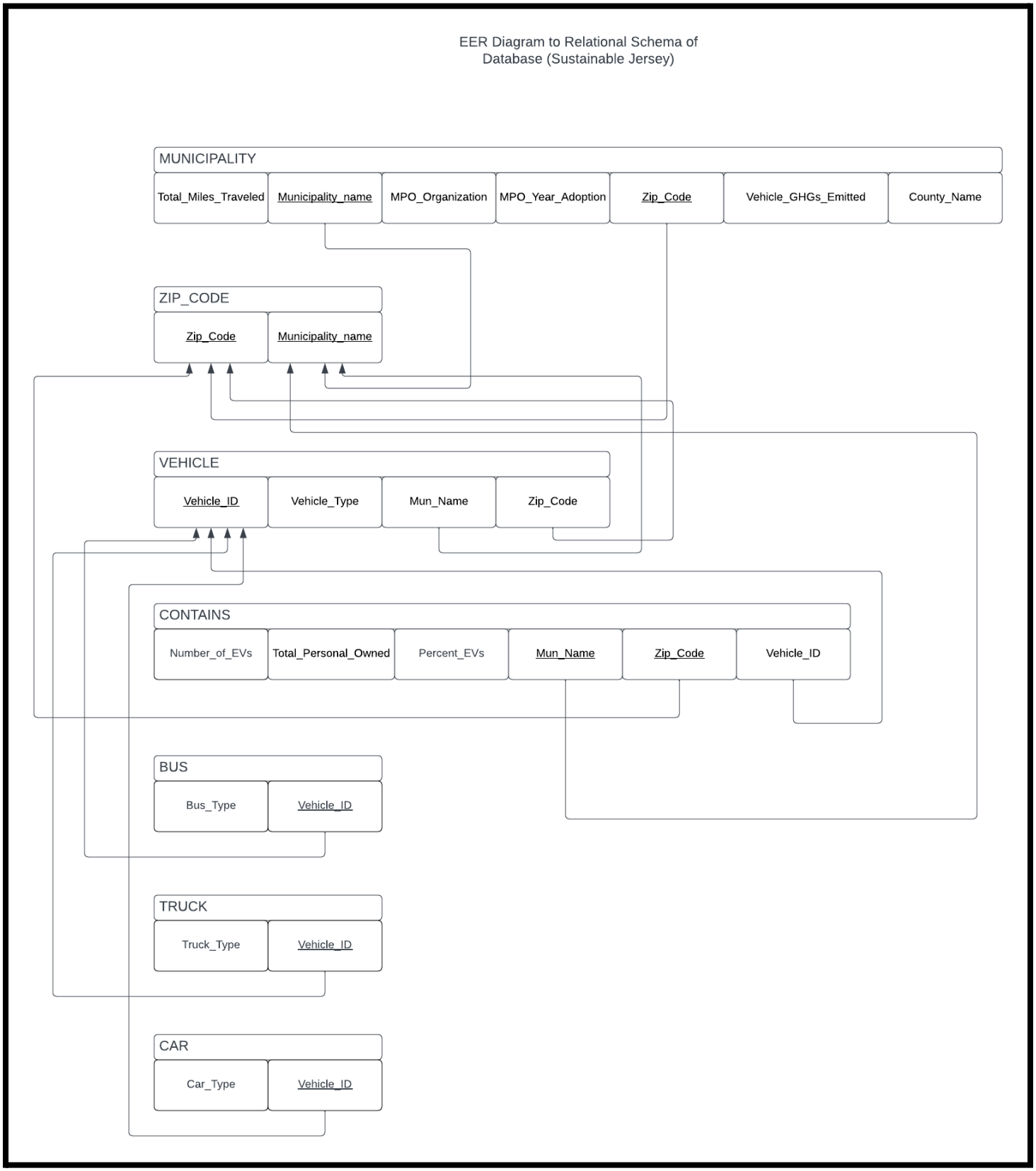
Phase III – ER to Schema & Expected Data Size
Madison Bavosa, Emmanuel Pasteur, EJ Gasataya

- ***Mapping of ER Diagram to Relational Schema***



Estimation of Database Size

File Name -> **usa_zipcode_of_nj-1548j.xlsx**

- Zip Code -> Can be an integer type -> 4 bytes
- Zip Code Name (Municipality) -> Assume max string length = 20 chars * 1byte =20 bytes
- City Name -> Assume max string length = 20 chars * 1byte =20 bytes
- State Initial (NJ) -> 2 Characters * 1 byte = 2 bytes
- County Name -> Assume max string length = 15 chars * 1 byte = 15 bytes

- Total Number of entries in Excel sheet -> 705 entries (excluding row for col. name and info)
- $Rough\ Size: \ 705 \ * \ (4 \ + \ 20 \ + \ 20 \ + \ 2 \ + \ 15) \ \approx \ 43005 \ bytes$

File Name -> **Electric_Vehicle_Ownership_Data_06.01.22.xlsx**

- Municipality -> Assume max string length = 20 chars * 1byte= 20 bytes
- County Name -> Assume max string length = 15 chars * 1 byte = 15 bytes
- Year -> Can be an integer type = 4 bytes
- Total # of Personal Vehicles -> Can be an integer type = 4 bytes
- Total # of EVs -> Can be an integer type = 4 bytes
- %age of EVS -> Can be a double type = 8 bytes

- Total Number of entries in Excel sheet -> 1130 entries (excluding row for col. name and info)
- $Rough\ Size: \ 1130 \ * \ (20 \ + \ 15 \ + \ 4 \ + \ 4 \ + \ 4 \ + \ 8) \ \approx \ 62150 \ bytes$

File Name -> **Vehicle_Miles_Traveled-On-Road_Vehicle_GHG_Emissions_Data_08.11.22**

- **(This is the VMT Data Section)**
- Municipality -> Assume max string length = 20 chars * 1byte= 20 bytes
- County Name -> Assume max string length = 15 chars * 1 byte = 15 bytes
- MPO -> String length is 5 = 5 chars * 1byte = 5 bytes
- Year -> Can be an integer type = 4 bytes
- Note: The following columns after year contain ***all*** **integer types** -> 14 * 4 bytes = 56 bytes

- Total Number of entries in Excel sheet -> 1131 entries (excluding row for col. name and info)
- $Rough\ Size: \ 1131 \ * \ (20 \ + \ 15 \ + \ 5 \ + \ 4 \ + \ 56) \ \approx \ 113100 \ bytes$

File Name -> **Vehicle_Miles_Traveled-On-Road_Vehicle_GHG_Emissions_Data_08.11.22**

- **(This is the GHG Emissions Data Section)**
- Municipality -> Assume max string length = 20 chars * 1byte= 20 bytes
- County Name -> Assume max string length = 15 chars * 1 byte = 15 bytes
- MPO -> String length is 5 = 5 chars * 1 byte = 5 bytes
- Year -> Can be an integer type = 4 bytes
- Note: The following columns after year contain ***all*** **double types** -> 14 * 8 bytes = 112 bytes

- Total Number of entries in Excel sheet -> 1131 entries (excluding row for col. name and info)
- $Rough\ Size: \ 1131 \ * \ (20 \ + \ 15 \ + \ 5 \ + \ 4 \ + \ 112) \ \approx 176436 \ bytes$

The types of searches describes the complexity of your queries; make a rough estimate of the number of searches that your algorithm will require.

- Since our schema will roughly contain 8 possible relations (as shown in our Relational Schema Mapping), we would expect our queries to be more complex.

- Since calculating the EV percentage can be calculated based on the *ratio* (division) of the total number of personal vehicles by the total number of EVs in *each* municipality, we suspect that the number of searches will roughly equal the number of rows where each municipality is listed; because each municipality will have a different ratio of EV/non-EV usage.

- The sample size in all of our datasets will also have to be taken into account (as mentioned above for database size), which may influence the performance of complex queries.

- In our proposal, our goal was to specify which municipalities have certain EV usage based on Zip-Code, and to also specify the list of municipalities that fall *within the GHG emission range we specify*, and relate the two results together. Thus the complexity of the query will also depend on which ranges the user might enter. The aggregate functions “MIN and MAX” that we have learned in relational algebra could allow the database to restrain which tuples to filter. (ex. Apply the aggregate functions on Vehicle_GHGs_Emitted in the “MUNICIPALITY” relation)

- In the explanation of our database model, we introduced another use case in which users will be able to enter a mileage range and the database will return the list of municipalities that fall within the range. This would require the use of the aggregate functions “MIN” and “MAX” on the “Total_Miles_Traveled” attribute on the “MUNICIPALITY” relation. The range will determine the number of searches the database will require.