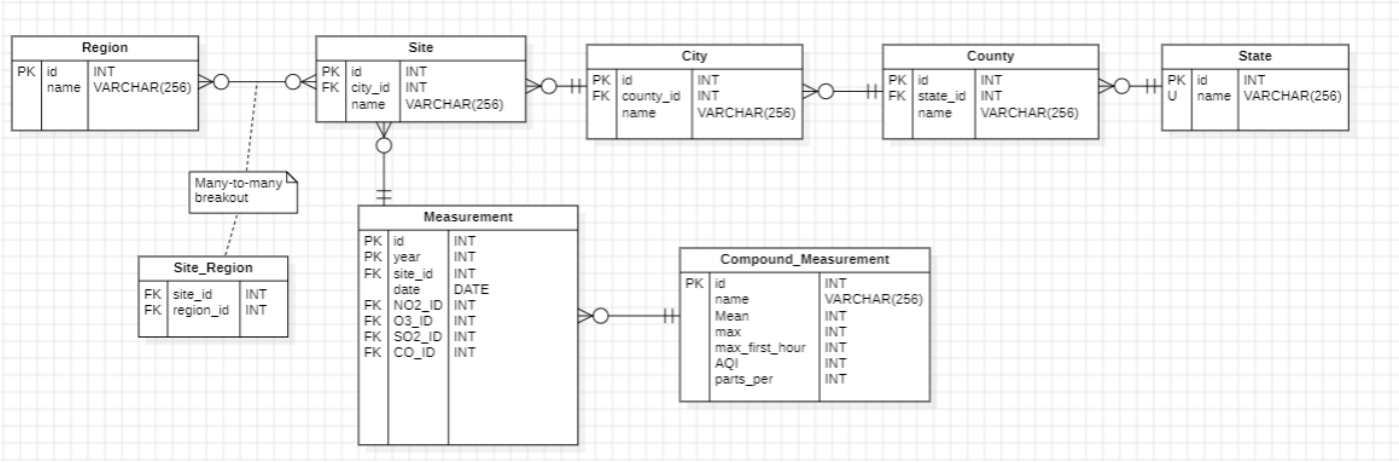
Project Stage 2

UML Diagram



Assumptions

- This schema assumes that the data_ID found in column A of the provided dataset will always loop back to zero only upon the change of the year.
- New compounds will only be added by adding a new ID attribute to the schema of the Measurements table.
- This system was specifically designed to make as few assumptions as possible, assuming that it would be possible to add new sites to the data, and that those sites would not inherently be added to the end of the list.

Normalization Rationale

We chose to apply the Third Normal Form method of normalization for this table. This method was selected to prioritize database integrity and reduce the total storage capacity used by redundant strings. Because every string is only stored once in their respective entities, data anomalies and discrepancies would be minimized with this model. The 3NF form also conveniently eliminates the need to use strings or any non-integer data type as a primary key and perform the minimal amount of comparisons between strings. As designers, we chose to prioritize eliminating update, deletion and insertion anomalies over slight performance downgrades. It is our understanding that this mentality is preferred in industry when handling larger, more complicated, and critical data.

Normalizing a table to the 2nd Normal Form concerns eliminating sub-optimal functional dependencies. Every relation with a primary key that is only a single attribute is automatically 2nd Normal Form. Therefore, the only required examination is for the “Measurements” table. Each dependent attribute in the measurement table is entirely defined by both the Data_ID and Year. Only having one of these attributes would produce a myriad of conflicting results for every dependent attribute. Therefore, that table is also in 2nd Normal Form.

Normalizing from 2NF to 3NF primarily targets eliminating redundancies. The main goal is to reduce or remove the potential for update anomalies. This means removing “transitive dependencies” For example, the state that a measurement was taken in is dependent on the county that the measurement was taken in. Therefore, we expanded the schema of our database to expand the geographical information into separate tables. While mistakes such as incorrectly making “Orange County, California” as “Orange County, Florida” are still possible, we considered eliminating the possibility of an “Oroonge County, California” to be more pertinent.

As the UML Diagram stands, every functional dependency is dependent on the entire primary key, and nothing but the entire primary key. This includes the elimination of transitive dependencies. Therefore, the database is normalized in 3NF Form.

Relational Schema

Region(id:INT [PK], name:VARCHAR(256))

Site(id:INT [PK], city_id:INT [FK to City.id], name:VARCHAR(256))

City(id:INT [PK], county_id:INT [FK to County.id], name:VARCHAR(256))

County(id:INT [PK], state_id:INT [FK to State.id], name:VARCHAR(256))

State(id:INT [PK], name:VARCHAR(256))

Site_Region(site_id:INT [FK to Site.id], region_id:INT [FK to Region.id])

Measurement(id:INT [PK],
year:INT,
site_id:INT [FK to Site.id],
NO2_id:INT [FK to Compound.id],
O3_id:INT [FK to Compound.id],
SO2_id:INT [FK to Compound.id],
NCO_id:INT [FK to Compound.id],
date:DATE
)

Compound_Measurement(id:INT [PK],
name:VARCHAR(256)
mean:FLOAT,
max:FLOAT,
max_first_hour:FLOAT,
AQI:FLOAT,
parts_per:FLOAT)