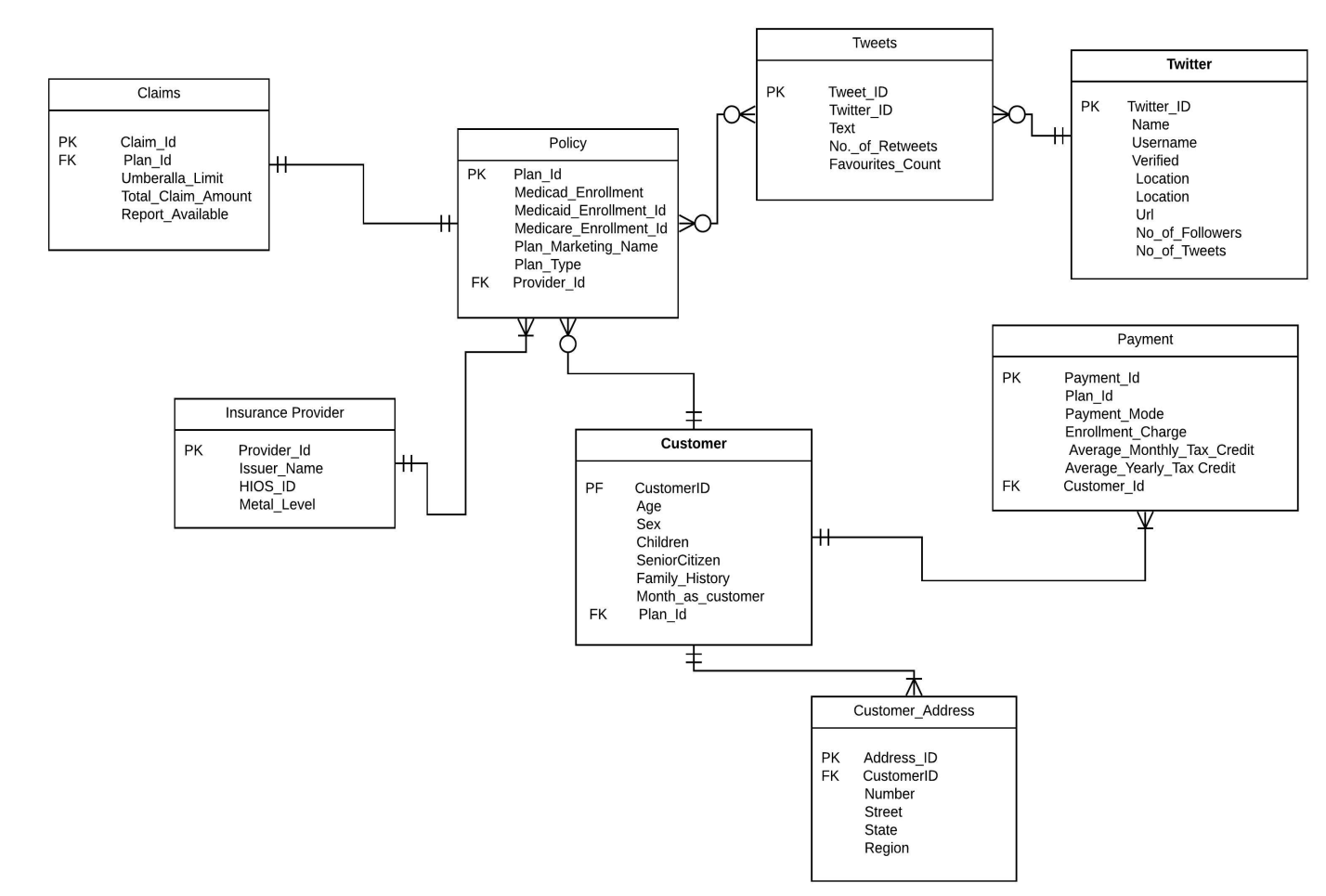
**README**

**Files used:**

insurancedata.csv

The CSV file that was used in this project is obtained from kaggle and Scrapping Social Media for getting masked and real-time dataset. To work with this project we need to have an active Twitter developer account which is used to generate the Twitter API from which we scrape the tweets that are needed for our database.



The ER diagram we used here is used to model the entire database of our project.

**Relationship:**

Each of the tables relationship is as follows:

***Customer - Customer Address:***

A customer shall be having multiple address associated with them in real-world situation, hence we represent the tables linked as one-many relationship and a customer address attribute shall be associated with only one customer hence addressed by one -only -one relation.

***Customer - Policy:***

A customer shall have one policy taken under his name offered by one insurance provider or many policies depending on his/her requirement or in cases can choose not to take any policy too, hence we have represented by zero-many relationship. But a policy shall be associated with only one customer hence addressed by one -only -one relation.

***Policy - Insurance Provider:***

A policy can be offered by only one insurance provider hence we have the one-one relationship. But a insurance provider shall have multiple policies offered under their name hence one-many relationship.

***Policy - Claim:***

A claim is when a policy shall face claim status depending upon the customer needs. A claim shall always be associated with only one policy and same applies to vice versa hence both are represented by one-one relationship.

***Customer - Payment:***

When a customer takes in policy with the insurance provider there is generally multiple payments made to the provider for a single policy hence we have the one-many relationship. But always a payment made each time shall be associated to only one customer hence we have the one-one relationship.

***Provider - Twitter:***

A insurance provider shall choose to have one or many twitter user accounts to post tweets about any news on behalf of their company. Hence we have the one-many relationship. But a twitter id shall always be associated to only one company which is shown by one-one relationship.

***Twitter-Tweets:***

Here we analyze the tweet posted by each of the twitter users corresponding to our filter condition and analyze them. A tweet shall be posted by only one user hence the one-one relationship, but a user shall have multiple/no tweets posted by his id which is represented by the zero-many relationship.

**AWS:**

For our database setup we have chosen the AWS Cloud to hold the entire database. We have used the AWS RDS feature to create the setup infra needed to host the database. We have enabled the database to allow only specific connections to keep the access restricted.

