Database Info

The database file is sqlite/db/Fintech.db

The code dealing with the database is in three different files: DataPull/sql\_functions.py, fintech\_django.sql\_functions.py, and sqlite/sqlite\_script/main.py

DataPull/sql\_functions.py contains the functions for adding info to the database and converting info from the database into a csv file for predictions, the functions are as follows.

* add\_history(datetime, str, float, float, float, float) – this function adds info into the database. The info it adds is a date, company name, and three prices. The prices are one day before, one day after, one month after, and three months after.
* convert\_date(datetime) – this turns the date into an int the database can use.
* output\_to\_csv() – this turns all the history in the database into a csv file formatted in colums as follows: company name, date, day before price, day after, month after, three months after, transcript.
* get\_companies\_help() – this function returns an array of the different companies in the database.
* add\_prediction\_short(str, float) – add\_prediction\_medium() – add\_prediction\_long() – all three of these functions accept a company name and prediction and then add them to the associated table.

fintech\_django/sql\_functions.py contains the functions the frontend uses to receive info from the database, the functions are as follows.

* get\_companies\_help() – the same function as used in DataPull/sql\_functions.py
* get\_companies() – adds the array from get\_companies\_help() and puts it in to a dictionary, turns it into json and returns the json.
* get\_history(datetime, datetime, str) – takes two dates and the company name and returns the json of a dictionary which holds the array of tuples, where each tuple is a row from the table pertaining to the company.
* get\_prediction\_short(str) - get\_prediction\_medium(str) - get\_prediction\_long(str) – each of these return the json of a dictionary of an array of tuples (the array should only have a length of one if the database is working correctly) where the tuple is the company name and prediction from the associated table.
* convert\_date(datetime) – the same function as in DataPull/sql\_functions.py
* get\_current(str) – returns the most recent day before price in the table associated with the company name.

sqlite/sqlite\_script/main.py is a command line program to run for developer convenience. The program can be run with different commands as follows.

* -h or -help – these return the list of possible commands
* -c – this will create a table if it doesn’t already exist. Will run for each name after -c
* -d – this will instead drop a table if it does exits, Will run for each name after -d
* -s – this will display all tuples from a table, will run for each name after -s
* -n – this provides a list of all tables/names of companies in the database
* -cs – this creates the short prediction table
* -cm – this creates the medium prediction table
* -cl – this creates the long prediction table
* -ss – this displays all tuples from the short prediction table
* -sm – this displays all tuples from the medium prediction table
* -sl – this displays all tuples from the long prediction table
* -ct – this displays the count of all tuples in a table, will run for each name after -ct
* -dd – this will delete any dummy data from all table (dummy data is classified as tuples where the transcript is ‘transcript’)