

Dear Candidate,

We would like you to complete a case study. This will help us assess your coding skills, see how you work with and think about data, and give you a better understanding of the work you could expect to do at Prism.

There is a data.txt file attached containing a single day's worth of tick data for options on the global technology ETF (Exchange Traded Fund) IXN. Each line in the file is called a "message". There are 5 message types:

- S: Static
- Q: Quotes
- C: Close
- T: Trade
- R: Recap

The type of each message can be determined by the 3rd field in each line. This is an example of a static message:

```
15-04-22|00:00:08:285|S|IXN22EKB61500|126|2|1407801616570000200|f1=2|f2=IXN|f4=20|
```

Each message has the following format:

```
Date|timestamp|message type|instrument code|exchange code|security type|unknown|field1|field2|....
```

So, the instrument code for the message above is IXN22EKB61500, the exchange code is 126, and the security type is 2. Each field takes the form *f*<field_id>=<field_value>. The StaticFields.txt and DynamicFields.txt files provide a mapping between the <field_id> and its name for static fields and dynamic fields, respectively (all fields under Q, C, T, and R messages are considered dynamic).

Your task is to write a series of Python scripts that will allow a user to request data for a single instrument code around a given timestamp. We ask you to do this in 4 parts:

1. Write a python script which splits the raw data into 5 csv files, one for each message type. The headers of the field columns should be taken from the DynamicFields.txt or StaticFields.txt files, and the values should only contain the <field_value> (e.g., 20 instead of f4=20). For other columns you can use the labels used in the example above. You can drop any columns which do not have a field name. You do not need to include date or time information for static data.
2. Design a set of SQLite tables to store data for each of the files produced in step 1. We have provided you with a skeleton project to do this.
3. Write a python script to populate your database using the output of step 1.
4. Write a python script which takes a time and instrument code as an input and prints a table of all quote data within 10 minutes before and after chosen time across all dates. You should join these results with any available static data. An example input would be 15:00:00:000 and IXN23MKB57500. The output should be a table with all quote data for this instrument obtained between 14:50:00:000 and 15:10:00:000, joined with any available static data.

Please ensure the code is checked-in to your Github account and contains a short readme explaining how to run it locally. Please make sure the Github repository is private.

Good luck!