CMPUT 291

Mini Project 2

2019-11-24

Abstract:

The project converts email records provided in an XML format to text files of specified formats. It then converts these files into index files from which are later queried to produce the desired output.

Running Instructions:

Phase 1: To run Phase 1's code, simply call the python file named *phase1.py* with the command line argument being the XML file to create the files *terms.txt*, *emails.txt*, *dates.txt*, and *recs.txt* from. This will produce these files in the same directory as *phase1.*py is located. Example:

```
>> python phase1.py thousand.xml
```

Phase 2: To run Phase 2's code, simply call the python file named *phase2.py*. This will create intermediate files and the index files, being *re.idx*, *te.idx*, *em.idx*, and *da.idx*. Example:

```
>> python phase2.py
```

Phase 3: To run Phase 3's code, and bring up the query interface, simply call the python file named *phase3*.py. Follow the instructions on screen or write queries given the query language. Example:

```
>> python phase3.py
```

Testing Strategy:

Phase 1: Phase 1 was tested by creating the *phase1.py* file and then comparing its output to the desired output provided on eClass for both the 10 records and 1000 records. This was done using a diff tool to spot any differences between the files. If any difference was found, then we knew the file had an error that needed to be solved. Generally, this came down to formatting, or fixing a regular expression.

Phase 2: Phase 2 was tested by looking for sorted files and indexed files given the files had keys and data separated. Using *db_dump* in the terminal on the index file to see the indices, we were able to verify it was done properly.

Phase 3: To test Phase 3, the interface created was used, and a database was created according to the data retrieved specification. Queries were performed on data sets given.

Group Work Break-Down Strategy:

The work was divided approximately evenly, with Luke having solved Phase 1 earlier on during the week of development, having taken approximately 5 hours, and Aryan solving and starting Phases 2 and 3 later during the week having taken approximately 3 and 10 hours, respectively, for Phases 2 and 3. Phase 3 was later completed by both members of the team in the final days of development. Coordination was kept on track using messaging services like texting and source control was maintained using Git. This ensured both partners knew what each other were working on and had each other's code / work.

Additional Notes:

The queries containing wildcards do not work. Solutions were tested and did not produce correct results. The implementation was left in, despite not working.