CS384 2020 Assignment 3

Mayank Agarwal

14 Oct 2020

Solve between 15:00 hrs to 18::00 hours (3 hours)
Deadline: 23rd Sep 2020, 18:00 hrs
Note:

- You are allowed to use **csv, os, re** module.
- Usage of Pandas is **NOT** allowed
- Use of regular expression is allowed
- Creation of folders needs to be done via os module

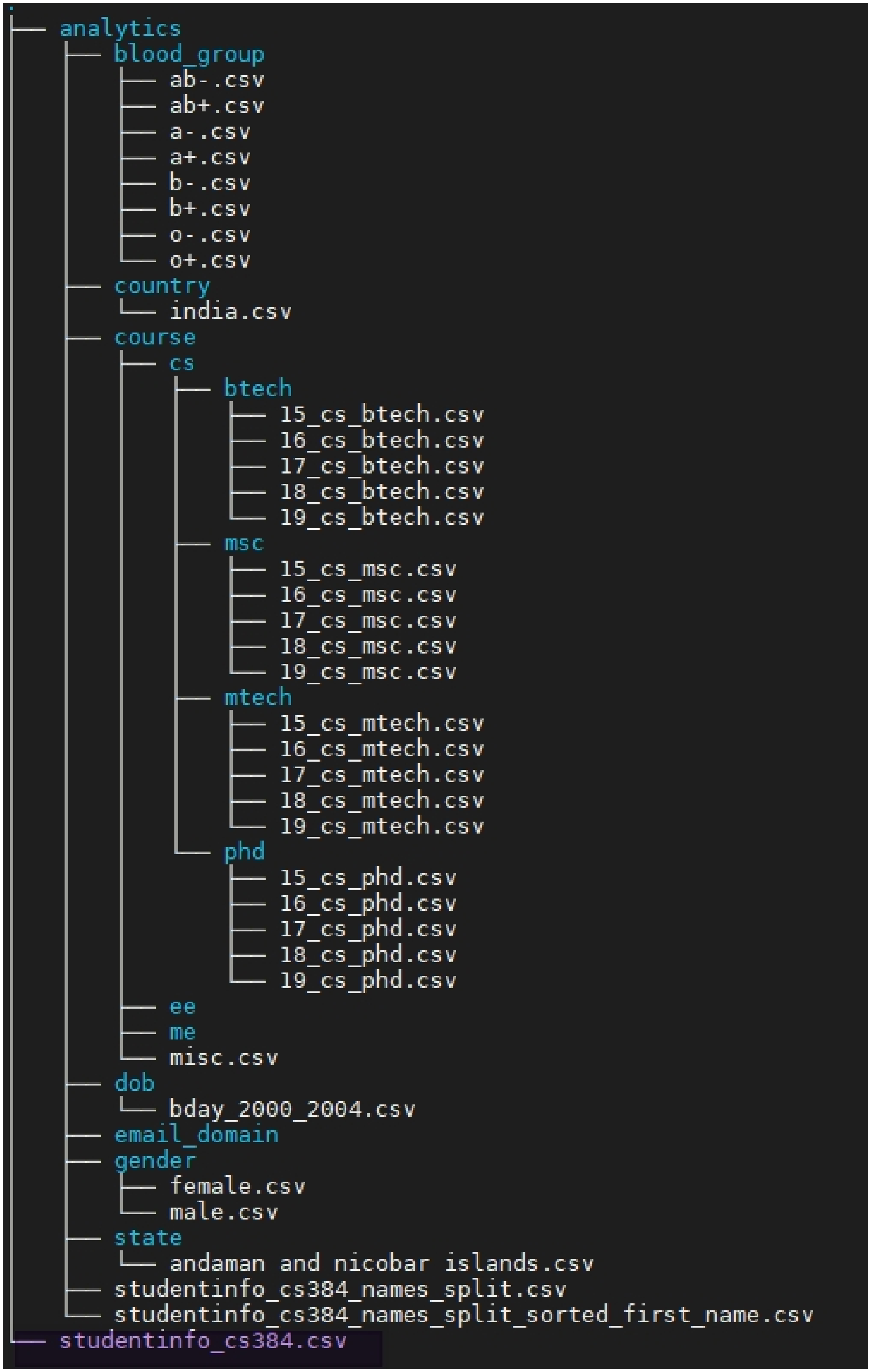
You are given a file "studentinfo_cs384.csv"

It contains the following fields. id,full_name,country,email,gender,dob Each field is self explanatory. The information is generated synthetically, however the roll numbers are real. As you know the IITP roll number has the following properties: For a given roll number 1701ME01

- 17 - Year of Admission (first two bits)
- 01/11/12/21 – Course - 01 Btech /11 Mtech /12 MSc/ 21 Phd (3rd, 4th bits)
- CS/ME/EE/PH etc.. two digit branch code (5th, 6th bit)
- serial number (last two bits)

Tasks: (Each task is a commit operation with at least 6 words, so for 'n' tasks, 'n' commits should be there, there can be more than 'n' commits, which is appreciated).

1. **Note**: In all the csv that you create all the cols of the file "studentinfo_cs384.csv" needs to be copied
2. Read the file "studentinfo_cs384.csv" using csv module
3. **Course**. Extract Roll number. For every roll number extracted. You need to make the following folder structure. I am showing for CS branch for other branches you need to make it automatically based on the scanned roll number. See Fig. 1. Dont make static defined paths. Assume that if a new roll number say 2001ME12 comes this time, that filtering should happen automatically and the directory structured needs to be created automatically. File names are (all lower case) $[2digit\_year]\_[2char\_branch\_code]\_[btech|msc|mtech|phd].csv$. All directory needs to be created using the os module and not manually by right clicking in the desktop OS.
4. **country** : for 'n' unique countries, you need to create 'n' csv files. For example, if the country is India, all those rows having country as India, should be in file, India.csv. Make other country files automatically by reading the country name from the csv.



```
.
├── analytics
│   ├── blood_group
│   │   ├── ab-.csv
│   │   ├── ab+.csv
│   │   ├── a-.csv
│   │   ├── a+.csv
│   │   ├── b-.csv
│   │   ├── b+.csv
│   │   ├── o-.csv
│   │   └── o+.csv
│   ├── country
│   │   └── india.csv
│   ├── course
│   │   ├── cs
│   │   │   ├── btech
│   │   │   │   ├── 15_cs_btech.csv
│   │   │   │   ├── 16_cs_btech.csv
│   │   │   │   ├── 17_cs_btech.csv
│   │   │   │   ├── 18_cs_btech.csv
│   │   │   │   └── 19_cs_btech.csv
│   │   │   ├── msc
│   │   │   │   ├── 15_cs_msc.csv
│   │   │   │   ├── 16_cs_msc.csv
│   │   │   │   ├── 17_cs_msc.csv
│   │   │   │   ├── 18_cs_msc.csv
│   │   │   │   └── 19_cs_msc.csv
│   │   │   ├── mtech
│   │   │   │   ├── 15_cs_mtech.csv
│   │   │   │   ├── 16_cs_mtech.csv
│   │   │   │   ├── 17_cs_mtech.csv
│   │   │   │   ├── 18_cs_mtech.csv
│   │   │   │   └── 19_cs_mtech.csv
│   │   │   └── phd
│   │   │       ├── 15_cs_phd.csv
│   │   │       ├── 16_cs_phd.csv
│   │   │       ├── 17_cs_phd.csv
│   │   │       ├── 18_cs_phd.csv
│   │   │       └── 19_cs_phd.csv
│   │   ├── ee
│   │   ├── me
│   │   └── misc.csv
│   ├── dob
│   │   └── bday_2000_2004.csv
│   ├── email_domain
│   ├── gender
│   │   ├── female.csv
│   │   └── male.csv
│   ├── state
│   │   └── andaman and nicobar islands.csv
│   ├── studentinfo_cs384_names_split.csv
│   └── studentinfo_cs384_names_split_sorted_first_name.csv
└── studentinfo_cs384.csv
```

Figure 1: Directory Structure. Its shown partially only, but needs to auto populated for other cases too. "studentinfo_cs384.csv" is the source file.

5. **email**: extract the domain from the email address. In this case, extract the first word after @, so for example gkindall5f@wunderground.com, the first first word after @ is wunderground. So the domain is wunderground. Similary in akeane5o@timesonline.co.uk the domain is timesonline. So for all domains,

you need to make a domain.csv. for ‘n’ unique domains, you need to create ‘n’ csv domains.

6. **gender**: two files. male.csv , female.csv

7. **dob**: the dates are in dd-mm-yyyy format. so you need to make 5 files after computing the year of birth. For those you are born between 01-01-1995 till 31-12-1999 , then 01-01-2000 till 31-12-2004, 01-01-2005 till 31-12-2009, 01-01-2010 till 31-12-2014, 01-01-2015 till 31-12-2020, bday_1995_1999.csv, bday_2000_2004.csv, bday_2005_2009.csv, bday_2010_2014.csv, bday_2015_2020.csv.

8. **blood_group**: make 8 files. Blood group file names need to be constructed by reading the col G from the “studentinfo_cs384.csv” and processing it. You cannot define your own static tuple/list.

9. **state**: for ‘n’ states, make ‘n’ csv. files. State names are to be extracted from the “studentinfo_cs384.csv” file state column. You cannot define your own static tuple/list. Just write a code to find the unique states from the file and create files accordingly. Again state names needs to be picked from csv and not static.

10. **new_file**: the input file has the following columns: id,full_name,country,email,gender,dob,blood_group,state

    make a new output file, “studentinfo_cs384_names_split.csv” having following cols
    id,first_name,last_name,country,email,gender,dob,blood_group,state

    Logic: split the name via space, and the left word [0] would be first name and the remaining [1:] will be last name .

11. **sort new_file (above file)**: Read the “studentinfo_cs384_names_split.csv” file and then sort by first_name and save as “studentinfo_cs384_names_split_sorted_first_name.csv” keeping all columns. Ensure header row (first row) is preserved.