MSDS 7330
File Organization and Database Management
Homework XML

Name: Peter Byrd

This is a homework assignment for MSDS 7330, File Organization and Database Management. For this assignment, turn in a single pdf file containing all of your answers. The file should be named ¡yourLastName¿HW-XML.pdf. For example, the file name for my homework assignment would be 'RafiqiMiniProject-XML.pdf'. Insert your answer pages into this file with the answer for Question 1 inserted immediately after Question 1 and before Question 2, the answer for Question 2 inserted immediately after Question 2, etc. You may insert a front page containing your name and date if you do not wish to or cannot electronically add that information to the first page of this homework sheet.

Collaboration is expected and encouraged; however, each student must hand in their own homework assignment. To the greatest extent possible, answers should not be copied but, instead, should be written in your own words. Copying answers from anywhere is plagiarism, this includes copying text directly from the textbook. Do not copy answers. Always use your own words and your own code. Directly under each question list all persons with whom you collaborated and list all resources used in arriving at your answer. Resources include but are not limited to the textbook used for this course, papers read on the topic, and Google search results. Don't forget to place your name on the first page of the pdf document.

XML Database

Question 1: The file baseball salaries 2003.txt contains salary information for certain professional baseball players from the year 2003. Define an XML schema for this file. Write a Python script that processes this file and stores it in a single XML file: baseball salaries 2003.xml.

Turn in the Python script, XML schema definition, and resulting XML file.

XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.example.org/XMLSchema">
<xs:element name="baseball" type="baseballType"/>
      <xs:element name="playerdata" type="xs.string">
            <xs:complexType>
                  <xs:sequence>
                        <xs:element name="team" type="xs:string"/>
                        <xs:element name="player" type="xs.string">
                        <xs:element name="salary" type="xs:integer"/>
                        <xs:element name="position" type="xs:string"/>
                  <xs:sequence>
            </xs:complexType>
      </xs:element>
</xs:schema>
```

Python Script:

```
# Mini Project 4
# Question 1

import pandas as pd
import xml.etree.ElementTree as ET


# read the player data text file into a python dataframe using panda
playerdata=pd.read_csv('/Users/pbyrd/anaconda/Data/Baseball/baseball_sala
ries_2003.txt',sep=":",skiprows=3,
names=('Team','Player','Salary','Position'))


# define a function that converts the dataframe to xml and writes to a
xml file
def df_to_xml(df, filename=None, mode='w'):
    def row_to_xml(row):
        xml = ['  <playerdata>']
        for i, col_name in enumerate(row.index):
            xml.append('    <{0}>{1}</{0}>'.format(col_name,
row.iloc[i]))
        xml.append('  </playerdata>')
        return '\n'.join(xml)
    res = '<?xml version="1.0" encoding="UTF-8"?>\n<baseball>\n' +
'\n'.join(df.apply(row_to_xml, axis=1)) + '\n</baseball>'

    if filename is None:
        return res
    with open(filename, mode) as f:
        f.write(res)


# execute the function on the dataframe 'playerdata', with output file
'baseball_salaries_2003.xml'
df_to_xml(playerdata,'/Users/pbyrd/anaconda/Data/Baseball/baseball_salari
es_2003.xml')
```

XML File (excerpt) – Full file attached in Canvas

```
<?xml version="1.0" encoding="UTF-8"?>
<baseball>
  <playerdata>
    <Team>New York Yankees </Team>
    <Player>Acevedo Juan  </Player>
    <Salary>900000</Salary>
    <Position> Pitcher</Position>
  </playerdata>
....
  <playerdata>
    <Team>Texas Rangers        </Team>
    <Player>Zimmerman Jeff     </Player>
    <Salary>3366667</Salary>
    <Position> Pitcher</Position>
  </playerdata>
</baseball>
```

Collaborators: **Harry Bhasin**

Resources: **https://stackoverflow.com**
https://docs.python.org/3/library/xml.etree.elementtree.html#tutorial

Question 2: The file baseball salaries 2003.xml contains salary information for certain professional baseball players from the year 2003. Write a Python script that processes the XML file from Question 1 to determine, for each position, the average salary of the players in that position. Note that the seven player positions that can occur in the input file are "Catcher", "First Baseman", "Outfielder", "Pitcher", "Second Baseman", "Shortstop" and "Third Baseman". The output should appear sorted in descending order of average salary.

Capture the resulting output in a file. Turn in a pdf of your script and the results.

Results:

```
outfielder 4050024.41
firstbase 3591402.63
shortstop 2953382.23
thirdbase 2461333.33
pitcher 2135130.19
secondbase 1307750
catcher 1172669.44
```

Python Script:

```
# Mini Project 4
# Question 2


import xml.etree.ElementTree as ET
import pandas as pd
from statistics import mean


# read xml file into Python
filename='/Users/pbyrd/anaconda/Data/Baseball/baseball_salaries_2003.xml'
root = ET.parse(filename).getroot()

# seperate data into positions
catcher = [playerdata for playerdata in root.findall('playerdata') if
playerdata.findtext('Position') == ' Catcher']
firstbase = [playerdata for playerdata in root.findall('playerdata') if
playerdata.findtext('Position') == ' First Baseman']
outfielder = [playerdata for playerdata in root.findall('playerdata') if
playerdata.findtext('Position') == ' Outfielder']
pitcher = [playerdata for playerdata in root.findall('playerdata') if
playerdata.findtext('Position') == ' Pitcher']
secondbase = [playerdata for playerdata in root.findall('playerdata') if
playerdata.findtext('Position') == ' Second Baseman']
shortstop = [playerdata for playerdata in root.findall('playerdata') if
playerdata.findtext('Position') == ' Shortstop']
thirdbase = [playerdata for playerdata in root.findall('playerdata') if
playerdata.findtext('Position') == ' Third Baseman']
```

```
# define a function to determine the mean salary for each position
def avg_salary(position, filename):
    salaryvector = []
    for playerdata in filename:
        salary = int(playerdata.findtext('Salary'))
        salaryvector.append(salary)
    avg_salary = mean(salaryvector)
    res = print(position, round(avg_salary,2))
    return res

# run the function for each position
avg_salary('outfielder',outfielder)
avg_salary('firstbase',firstbase)
avg_salary('shortstop',shortstop)
avg_salary('thirdbase',thirdbase)
avg_salary('pitcher',pitcher)
avg_salary('secondbase',secondbase)
avg_salary('catcher',catcher)

# sort the results
#res.sort(['Salary'], ascending=False)
```

Collaborators: **Harry Bhasin**

Resources: **https://docs.python.org/3/library/xml.etree.elementtree.html#tutorial**