# Lab 6 DBMS

106119029

## Question 1: Getting familiar with XML Files

Write a complete XML file named textbooks.xml, in which you describe at least a partial list of the textbooks you are using this semester. You should include at least two distinct textbooks. If you are using only one text this semester, expand your list to cover the current academic year. Your description of each book must include the title, author(s), publisher, year of publication, and the ISBN. For each author, specify the first and last name as distinct values. Include both a name and website for the publisher.specify the edition and cover type (paperback or hardcover) of each book you are using. Make sure your final XML document is well-formed.


## Question 2

Write a python program to read the following XML file (Given below)

1. Read the score of each class from `score_data.xml`, compute the GPA.
2. Add a gpa attribute for each class element.
3. Write the updated xml into a new file: output.xml (shown in Figure 2)
4. Upload your python source code file.

- `score_data.xml`

```
<?xml version="1.0"?>
<score_data>
    <student student_id="A001">
        <xml_class>60</xml_class>
        <data_structure>70</data_structure>
        <algorithm>85</algorithm>
        <network>90</network>
    </student>
    <student student_id="A002">
        <xml_class>66</xml_class>
        <data_structure>78</data_structure>
        <algorithm>62</algorithm>
        <network>88</network>
    </student>
    <student student_id="A003">
        <xml_class>89</xml_class>
        <data_structure>77</data_structure>
        <algorithm>80</algorithm>
        <network>50</network>
    </student>
</score_data >
```

### Python file

- q2.py

```
import xml.etree.ElementTree as et
arr = [0, 50, 60, 63, 67, 70, 73, 77, 80, 85, 90, 101]
ranges = list(zip(arr[:-1], [x-1 for x in arr[1:]]))
grades = [0, 1, 2, 2.3, 2.5, 2.7, 3, 3.3, 3.7, 4, 4.5]
rev_mapper = {k: range(v[0], v[1]+1) for k, v in zip(grades, ranges)}
mapper = dict([item for sublist in [[(x, k) for x in v]
                                    for k, v in rev_mapper.items()] for item in sublist])
tree = et.parse("score_data.xml")
print(tree)
root = tree.getroot()
#  Updating the XML
for elem_lv_1 in root:
    print(elem_lv_1.attrib)
    for elem_lv_2 in elem_lv_1:
        print("\t", elem_lv_2.tag, elem_lv_2.attrib, elem_lv_2.text)
        elem_lv_2.set("gpa", str(mapper[int(elem_lv_2.text)]))

# Converting it to string format
updated_xml = "<?xml version=\"1.0\"?>\n" + et.tostring(root).decode("utf-8")

# Writing it to a file
myfile = open("output.xml", "w")
myfile.write(updated_xml)
myfile.close()
```

- output.xml

```
<?xml version="1.0"?>
<score_data>
    <student student_id="A001">
        <xml_class gpa="2">60</xml_class>
        <data_structure gpa="2.7">70</data_structure>
        <algorithm gpa="4">85</algorithm>
        <network gpa="4.5">90</network>
    </student>
    <student student_id="A002">
        <xml_class gpa="2.3">66</xml_class>
        <data_structure gpa="3.3">78</data_structure>
        <algorithm gpa="2">62</algorithm>
        <network gpa="4">88</network>
    </student>
    <student student_id="A003">
        <xml_class gpa="4">89</xml_class>
```

```
        <data_structure gpa="3.3">77</data_structure>
        <algorithm gpa="3.7">80</algorithm>
        <network gpa="1">50</network>
    </student>
</score_data>
```

## Question 3

Write a python program to read the XML file output.xml (given below).

1. Add 5 to each class's score, compute the GPA.
2. Compute the average GPA.
3. Add average element for average GPA of each student
4. Write the updated xml into a new file: new_output.xml

### Python file

- q3.py

```
import xml.etree.ElementTree as Et

arr = [0, 50, 60, 63, 67, 70, 73, 77, 80, 85, 90, 101]
ranges = list(zip(arr[:-1], [x-1 for x in arr[1:]]))
grades = [0, 1, 2, 2.3, 2.5, 2.7, 3, 3.3, 3.7, 4, 4.5]
rev_mapper = {k: range(v[0], v[1]+1) for k, v in zip(grades, ranges)}
mapper = dict([item for sublist in [[(x, k) for x in v]
                                    for k, v in rev_mapper.items()] for item in sublist])


def main():
    tree = Et.parse('output.xml')
    root = tree.getroot()
    # Iterate through each element of root
    for sub_root in root:
        avg = 0.0
        count = 0
        temp = sub_root
        # Iterate over each sub-element
        for sub_sub_root in sub_root:
            # update marks
            sub_sub_root.text = str(int(sub_sub_root.text) + 5)
            # recalculate gpa
            gpa = mapper[int(sub_sub_root.text)]

            sub_sub_root.attrib['gpa'] = str(gpa)
            count += 1
            avg += gpa
```

```
            temp = sub_sub_root
        avg /= count
        temp.tail = "\n\t\t"
        c = Et.Element("average")
        c.text = str(avg)
        c.tail = "\n\t"
        sub_root.insert(count, c)

    # save updated xml to new_updated.xml
    tree.write('new_output.xml', encoding='utf-8', xml_declaration=True)


main()
```

- new_output.xml

```
<?xml version='1.0' encoding='utf-8'?>
<score_data>
    <student student_id="A001">
        <xml_class gpa="2.3">65</xml_class>
        <data_structure gpa="3">75</data_structure>
        <algorithm gpa="4.5">90</algorithm>
        <network gpa="4.5">95</network>
        <average>3.575</average>
    </student>
    <student student_id="A002">
        <xml_class gpa="2.7">71</xml_class>
        <data_structure gpa="3.7">83</data_structure>
        <algorithm gpa="2.5">67</algorithm>
        <network gpa="4.5">93</network>
        <average>3.35</average>
    </student>
    <student student_id="A003">
        <xml_class gpa="4.5">94</xml_class>
        <data_structure gpa="3.7">82</data_structure>
        <algorithm gpa="4">85</algorithm>
        <network gpa="1">55</network>
        <average>3.3</average>
    </student>
</score_data>
```