Assignment Three – XML

Weight: 10%   **Deadline: 15:30 (BST) 1 April**

Overview

The coursework takes the form of a group project. You should form a group of 3 and register this on Vision using the group sign up form linked from the Coursework 3 Assessment section.

The coursework involves the implementation of an XML data file, definition of the associated schema, and querying the XML data. To get a good mark, your coursework must demonstrate knowledge of the hierarchical semi-structured nature of XML, i.e. do not directly encode tabular data.

Groupwork

Share the work among the group members, ensuring that everyone has an equal share. All members should contribute to all areas of the coursework as experience gained in these topics will help in the examination.

Ensure that all members of the group have access to the files developed by the group. You must ensure that it is secure against other groups gaining access.

Please provide a summary in the report stating the contributions of each group member. If some group members do not participate, there is no need to generate queries on their behalf. Marks may be adjusted based on effort reported.

Collaboration and Plagiarism

Coursework reports and code must be the group's own work. If some text or code in the coursework has been taken from other sources, these sources must be properly referenced. Failure to reference work that has been obtained from other sources or to copy the words and/or code of another student is plagiarism[1] and if detected, this will be reported to the School's Discipline Committee. If a group is found guilty of plagiarism, the penalty could involve voiding the course.

Students must never give hard or soft copies of their coursework reports or code to students in another group. Students must always refuse any request from another student not in their group for a copy of their report and/or code. It is expected that all group members will have read and write access to the report and code for their group.

Sharing a coursework report and/or code with another group is collusion, and if detected, this will be reported to the School's Discipline Committee. If found guilty of collusion, the penalty could involve voiding the course.

---

[1] Heriot-Watt guidelines on plagiarism
https://www.hw.ac.uk/students/studies/examinations/plagiarism.htm

Scenario

HW Motors have a range of cars and vans for hire from a range of locations within the UK. For each hire, they need to store details about:

- The vehicle that has been hired including the dates of the hire, location of start and end of the hire, the hire category (small car, medium car, family car, MPV, sports car, luxury car, and minivan), and whether the vehicle is an automatic or manual.
- The person renting the vehicle including their driving license details, name, date of birth, home address, phone numbers (any combination of home, mobile, and office), and email address when available
- The fixed hire price per day.

Tasks

1. Design an XML file to represent data from the scenario defined above. When modelling your data **think about the XML semi-structured hierarchical model** *as opposed to the relational model covered earlier in the course.*

   Create a file with sample data. Choose suitable elements and enough data to enable you to provide a range of XPath expressions in the 3rd section below.

   Your XML data must be in a file named group01.xml, where you replace 01 with your group's number, and submitted in the archive submission.

   (5 marks)

2. Specify an XML schema to describe the data in your XML file. Show your knowledge of XML Schema constructs such as custom defined types. When developing your schema, ensure that your XML file validates against it. You can use the xmllint command line tool to validate[2] your file with the following command:

   ```
   xmllint --schema group01.xsd group01.xml
   ```
   where group01.xsl is the name of the file containing the XML Schema and group01.xml is the file containing the XML data.

   You XML Schema must be in a file named group01.xsd, where you replace 01 with your group's number, and submitted in the archive submission.

   (5   marks)

3. Create two XPath expressions per group member to obtain useful information from your XML file. Each XPath query must be distinctive from the others in your group, i.e. not just changing a parameter value.

   Each query should be saved in a separate file using the pattern group01ajg33-1.xpath, where 01 should be replaced by your group number, ajg33 should be replaced by the username of the group member whose query it is, and then a 1 or a 2. You can evaluate[3] your query using the xpath script available on Vision using the following command

   ```
   xpath.sh group01.xml group01ajg33-1.xpath
   ```

   (6 marks)

---

[2] You can also use the following online service to validate the XML file against your XML Schema. http://www.xpathtester.com/validate

[3] You can also use XPath checking tool http://www.xpathtester.com/xpath

Submission Instructions

The assignment is to be submitted through the two assignments on Vision.

1. Report through TurnItIn assignment. The report should contain:
   - The names of the people in the group.
   - Details of how the work was divided between the group members.
   - An overview in English of the contents of the XML file.
   - For each XPath expression, a description in English, the expression, and the results (copied from running the query over the data). It should be made clear which group member has contributed which query

2. An archive file as zip or tar.gz (no other archive formats will be marked) through the Grade Centre assignment. The archive file should contain:
   - 1 XML file containing your data named group01.xml
   - 1 XMLSchema file named group01.xsd
   - 2 XPath files per group member named group01ajg33-1.xpath

Hints and Tips

- The most difficult bit is probably validating the XML file against the XML schema file – allow plenty of time.
- Use a code editor NOT Word or Open Office Writer to create your XML files – you may get strange whitespace characters
- Don't leave any blank lines – (UNLIKE programming when blank spaces aid readability, they are invalid in XML) – if in doubt, remove other blank spaces.
- Don't use reserved XSL words like 'type' for your own element names.
- Write your own XML Schema rather than using a tool.
  The tool versions that I have seen deduce the list of suitable elements from what you have provided, e.g. limiting the list of countries or suitable dates to only those in your XML file. While this sometimes makes sense (e.g. maybe you have <gender>M</gender>, <gender>F</gender>, and <gender>N</gender> here a deduction that your only allowed values are M, F, and N is ok), but it doesn't make sense in those other situations when your XML file just contains example values (like dates), but you'd like to be able to store any date not just those listed in your example file. **In short, a tool is likely to provide an impressive-looking, over-complicated and incorrect version**.
- Watch out for simple errors like misspelling of element names.
- Other XML Schema validation and XPath evaluation tools exist and may be used for testing. Your files will be evaluated with the commands specified and must work with these tools.