

Assignment 5

Author: Rune Hjelsvold

Semester: Fall 2017

Last Update: 2017-10-30 20:20

Submission deadline: Sunday 5 November at 23:59

Objective

To practice XPath and DOM technologies and to get experience from storing information retrieved from an XML document into a database.

Please download the SkierLogs.xml file from the Blackboard page for Assignment 5.

Submission Requirements

Ensure that your submission fulfils these requirements:

- **Format:** Upload a short report, in PDF, to Blackboard. The report shall contain answers to the XPath questions, the UML model of your database, and the MySQL database schema. The report shall also include a reference to a Git repo where you keep the source files.
- **Naming Convention:** The report that you upload on Blackboard should be named *Assignment5_lastname_studentnumber.pdf*, using your own last name and student number.
- **Program Code:** Your program code should follow the coding conventions defined in http://www.tutorialspoint.com/php/php_coding_standard.htm. You MUST use the PHP DOM API when accessing the XML document (<http://php.net/manual/en/book.dom.php>).

Part I – Practicing XPath

This part will allow you to extend you XPath skills. Please run your answers in a browser to check that the expressions return what you expect when executed. Describe the results (number and type of nodes in the returned list when a node list is returned or numeric value returned when querying for `count()` and `sum()`).

Use the SkierLogs.xml file as the basis for these questions. Write XPath expressions that will

1. Find the fall years (the node list of `fallYear` attributes) for the seasons stored in the XML document.
Should return a list of two attribute nodes.
2. Find all the entries (the node list of `Entry` elements) logged by the user with user name `mari_dahl`.
Should return a list of 59 element nodes.
3. Find the logs (the node list of `Log` elements) for skiers in the 2015 season who skied for the club with id `vindil` and who skied more than 10 (kilometres).
Should return a list of five element nodes.

4. Find the user names (the node list of `userName` attributes) on skiers who in the 2016 season skied in an area having `Venabygd` as a part of the area name.
Should return a list of 17 attribute nodes.
5. Find number of skiers who are considered young juniors (i.e., they are born between 2002 and 2004).
Should return an integer value.
6. Find the dates (the node list of `Date` elements) during the season of 2016 where the skier with user name `idar_kals1` skied in the area named `Lygna`.
Should return a list of three element nodes.
7. Find the total distance logged during the 2015 season.
Should return an integer value.
8. Find the total distance logged in the season of 2015 by skiers who were not skiing for a given club in that season.
Should return an integer value.
9. Find the skiers in the skier list (the node list of `SkierLogs/Skiers/Skier` elements) who skied for an `Oppland` club in the season of 2015.
Should return a list 34 element nodes.
10. Find the skiers in the skier list (the node list of `SkierLogs/Skiers/Skier` elements) who skied in the area with the name `Nordseter` in the 2015 season but not in the 2016 season.
Should return a list of 5 element nodes.

Part II – Using DOM and PDO to Import XML Data to MySQL

In this part, you will get experience in using the DOM API to access XML data and in converting data from XML to a relational database.

Task 1 – Design the Database

Develop a conceptual (UML) model for the database. The database should store information about the skiers (unique user name, first name, last name, year of birth), about skiing clubs (unique id, club name, and city and county where the club is located). The database should also store information about what club (if any) the skier was representing a given season. It should be possible for join, move to or resign from a club between seasons – i.e., club affiliation may vary from one season to the other.

The database should also store the *total* distance that each skier logged each season.

Include the UML model in the report.

Task 2 – Create the Database in MySQL

Create tables, primary keys and foreign keys in MySQL for the database. Export the MySQL schema to an SQL file that you can import in the report and that you should store in your git repo too.

Task 3 – Importing XML Data to the Database

Develop a PHP script that opens the SkierLogs.xml file as a DOM Document. Use the DOM API to extract data about clubs and skiers from the XML file and use PDO to add records to the database for the clubs and the skiers. The script also needs to store what club (if any) the skier was representing in the various seasons. Finally, the script needs to sum up the logged distances for each skier for each season and store the season total for that skier to the database.

Hint:

It may be a good idea to design this as an M(V)C application. There is not need for any advanced view here since you are to demonstrate reading XML data, converting three-structured data to a relational form, and store it to a database. It makes sense to create a model consisting of a class that encapsulates the XML data, a class that encapsulates the relational database, and classes for in-memory representations of clubs and skiers and related data. The controller should first call the “XML model object” to convert the XML data to the in-memory representation and then call the “database model object” to store the in-memory data to the database.

There are two main benefits of this design:

1. A cleaner and easier to maintain program structure because details of the XML representation and the relational database representations are separated.
2. Easier to write and run unit tests because the conversion from XML can be tested independently from the code storing data in the database.