Alexandria University
Faculty of Engineering
Computer and Systems Engineering Dept.
Second Year
Fall 2016

CS221: Object Oriented Programming
Programming Assignment 3
Due: Tuesday, Nov. 29, 2016, 8:00 AM

Simple XML DBMS

Objectives

Upon completion of this assignment, you will be able to:

- Design an object oriented XML-based DBMS.
- Getting familiar with SQL and XML parsing.
- Draw a UML class diagram that represents your model.
- Apply different design patterns to your model.

Description:

A Computer Database is a structured collection of records or data that is stored in a computer system. On the other hand, a Database Management System (DBMS) is a complex set of software programs that controls the organization, storage, management, and retrieval of data in a database. DBMS are categorized according to their data structures or types. The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data. On the other hand, Extensible Markup Language (XML) is a set of rules for encoding documents in machine readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards.

Tasks:

- Implement a simple DBMS that handles data stored in XML files.

- You should support the following SQL Statements:
  - Create database
  - Create table
  - Insert into table
  - Delete from table
  - Drop database
  - Drop table
  - Select from table
  - Update table

- SQL is case insensitive.

- You should support the "`SELECT * FROM table;`" statement.

- You can find detailed SQL statements descriptions at: http://www.w3schools.com/sql

- For statements which contain conditions (i.e. the WHERE clause), support only the simple conditions: =, >, and <. You are NOT required to support multiple conditions: AND, OR, or NOT. (If you did, it will be bonus).

- You should create a DBMS Java interface and implement it. The DBMS interface should contain the functions that any program may need to use, for each of the supported statements.

  For example, to support database creation, you will add a method to the DBMS interface with signature similar to:

```
public void createDatabase(String databaseName);
```

  The DBMS interface does NOT handle SQL queries directly. SQL queries should be sent first to another class, say Parser class. The Parser parses them, and then calls the appropriate DBMS function. The Parser should use only the functions supported by the DBMS interface to access the database. The files should be updated, whenever any DBMS function (that requires updating) is called.

- The Parser should validate the SQL statements and reject bad ones.

- The DBMS should validate the sent parameters and should throw appropriate exceptions whenever needed.

- Although the code should be extendable enough, the table should support only two types for column values: “varchar” and “int”. “varchar” used to store string, and “int” to store numeric values (floating point is not required). Do not support custom type lengths (e.g. varchar(255), or int(11)), just assume all types of the same default length.

- You should provide a command line interface that accepts SQL queries.

- You are required to create a UML class diagram.

- Each table in the database should be stored in a separate XML file. Files should be placed in the database directory. You may use an environment variable to get the directory path during execution.

- You should maintain “Schema files” that contain data about the tables and columns. Tables should be validated across their schema files (“DTD Files” can be used for this purpose).

- This means each table will have at least two files: one for data (XML file), and one for its structure (DTD file).

- The DBMS you will develop should control the management and retrieval of data from data files. The DBMS accepts requests for data from application programs (in the form of the SQL queries) and retrieves and transfers the appropriate data from files that are stored physically on disk.

- You should use only SAX, DOM, or StAX parsers to parse and validate the XML database files.

General Code Requirements

- Your methods should not exceed 20 lines at any class (especially the main method).
- You should not use static variables/methods unless you really need them and have a reasonable justification.
- You should use Git to collaborate between different team members. Use bitbucket.org to create private repositories.
- You must use JUnit testing to test your functionalities, your testing code is required to be delivered too.

Deliverables

- You should work in groups of four. You are allowed to work in groups of five only if you could not find a group of three to join.
- Develop this assignment in Java.
- You should deliver your source code using your git repository.
- You should deliver a report that contains the required UML diagram, describes your design thoroughly, and contains a user guide that explains how to use your application. Any design decisions that you have made should be listed clearly.
- Delivering a copy will be severely penalized for both parties, so delivering nothing is much better than delivering a copy.