COP 4710/5716 **SQL and XML- Team Project** **Spring 2017**
100 Points (Project) + 20 Points (Presentation)

Due at the beginning of class (1:30 pm) on Monday, April 17.

As a database system designer, you will be working in a team to design and implement a temporal database for an SQL and XML software package in the data modeling class with the name TSQLx. The TSQLx package will be a valuable educational and research tool to help adding XML data similar to SQL data to a relational database with a temporal component for query purposes. The best team work from this semester will be selected to be used in future assignments in the data modeling course as well as other database courses. The work should include well designed software, documentation and a supplement of examples.

TSQLx will be implemented using C or Java in the UNF Osprey system by teams of 4-6 members who need to work closely together for the successful completion of the project. The software on Osprey will use a simple command line menu for each task such as create database, load database, create table, convert file to insert, insert to table, delete from table, query database tables, etc.

It is essential for each team member to be involved with her/his team on the database design, command structures, parser, programming, software presentation, and documentation. When you work as a team member in a real business organization, the delivery of a project on time, collaboration, participation, and cooperation are major components of a team member's responsibilities and the team's performance evaluation. As a team member in this project, it is essential to consider full participation in any of the components within the team project.

Structure for the Software
The data structure may support a structure similar to the diagrams given on the last page (Catalog Structure and Table Structure) and may be modified after team consultation with the instructor. The designed data structure should support a relational catalog and tables with attributes. The software should be written with a modular structure. The syntax for the commands should use a **parser** to provide appropriate messages for errors.

A. System Level Commands
Design the following system level commands for TSQLx.

a1. CREATE database command
The CREATE database will result in creation of the database catalog and structure for addition of the tables.

CREATE DATABASE database-name;

a2. DROP database command
The DROP database will remove the database from a system's external (permanent) memory.

DROP DATABASE database-name;

a3. SAVE database command
The SAVE command can be used after each system or user instruction for one or collection of commands to write the result in permanent memory.

SAVE DATABASE database-name;

a4. LOAD database command
The LOAD database command will load the database from external memory to internal memory.

LOAD DATABASE database-name;
When the database is loaded, the software could automatically capture system data/time or could ask for the **date and time** to be entered.

After creating or loading a named database the queries may take advantage of the **Commit** command to allow a queries (especially at DDL level) to be saved at database level.

B. Data Definition Language (DDL) -- (Will include Date and Time)

The commands at DDL for TSQLx should support create and drop with the following features similar to existing database systems.

To submit a DDL Query to the database, a user may utilize the **Commit** command to write the query to the database.

b1. CREATE table command

The CREATE table command should handle integer data, real data, and character data with size limitation. The size of the table(s) and columns of data should be determined by each team. The format of CREATE should match the following definition.

CREATE TABLE table-name (field-def [, field-def] ...);

where field-def in turn has the form
field-name field-type [NOT NULL]
field-type: integer[(n)], number[(w[, d])], character(n), **date (mm/dd/[yy]yy)**

Each table is created with an internal field that will store data/time for each record insertion. This date/time (as mentioned in section a4) captures the system data/time or the date and time that have been entered.

b2. DROP table command

The DROP table command should remove the table(s) from the database.

DROP TABLE table-name;

C. Data Manipulation Language (DML) with Data and Time Information

The software at the data manipulation language should handle the following commands.

c1. Standard Commands (will include Data and Time with each transaction)

INSERT INTO table [(field [,field]...)] VALUES (literal [, literal] ...);

Each insert command adds the system data/time or the date and time that has been entered to the internal field of the create command.

c2. Convert XML file to Insert Commands

The insert command also should provide another format that utilizes the supplied XML and XSD files to create one or more SQL insert command separated by semi-colon. The syntax for XML conversion command is shown below.

CONVERT XML filename[, XSD filename] as insertfilename;

The convert XML filename and XSD files should be parsed for structural syntax and correctness and then create the insert command(s). The parsed XSD file provides the attribute name, data type (such as string, number [integer, decimal], date/time and etc) and data size for database. XML file provides table, record, attribute names and attribute values for a database table(s).

c3. Insert commands for XML

To submit the XML insertfilename with insert command(s) created by the XML converter to the database, the following syntax can be used to submit each insert command in the file to the table.

INPUT insertfilename;

Note: Each insert as described in section c.1 will capture the system or entered data/time to write to the created field.

c4. Delete record(s) of a table

Similar to relational database commands, a delete may remove one record or multiple records from a table.

DELETE FROM table [WHERE condition];

c5. Simple display for content of a table.

Simple select will display the content of a table.

SELECT * FROM table; /** Normal select

Simple select with dates/time from a table with all attribute(s) listing

tSELECT * FROM table;

c6. Display for content of a table with attribute seletion

Selection of attributes and data/time from a table with or without condition

[t] Select [* | (field [,field]...)] FROM table [WHERE condition];

Required Materials to be Submitted

This project needs to be typewritten with well-defined reasoning for the data structure design and implementation. The final project should be submitted in class or right before start of the first presentation as a hard copy in flat folder and as electronic format on or before Monday, April 17 at 1:30 pm.

I. Hard copy in a flat folder containing different parts of the project
II. Electronic copy for hard copy on Blackboard
III. Electronic program modules as shar file on Osprey system for software testing

Required Materials to be Submitted in a Flat Folder
This project needs to be typewritten with well-defined reasoning for the data structure design and implementation and how to submit queries. The order of the material in the folder should follow the sequence given below.

I. Printouts
i1. Cover Page: Project name, name of the team members, course name, semester, submission date
i2. Description of assignment (maximum ½ page)
i3. Class presentation outline
i4a. Description of each team member's responsibilities in the TSQLx engine development, implementation and documentation
i4b. List of dates and times for meetings with topics discussed and team members in attendance

A separate **confidential evaluation** (sample distributed in class) of group member participation should be submitted to your

personal group account on Blackboard before midnight (11:59 pm) on Wednesday, April 19 to receive the grade for the confidential evaluation. The evaluation form will be provided to the class on Blackboard prior to the project due date. The confidential evaluation will be used to **determine each team member's level of participation in the group and his/her final project grade.** If a student does not provide confidential evaluation, it will be considered the student did not work on the project and a grade zero (0) will be assigned on the project.

i5. Description of the catalog and data structure with diagrams (link list -- support modularity) including any enhancements, how the solution has been approached, and the software designed by the team
i6. Description of the algorithm in pseudocode
i7. How to Run: Detailed instructions (step by step) about how to load files and run the program, how to submit queries, and how to quit the SQL prompt
i8. Program listing (list of the programs with routine or function name and comments on half page or less for each)
    -The source code should support meaningful variable names.
    -Comments should be written and embedded in the program (at necessary points, such as create, drop, select, etc. for routines and functions).
i9. Input and Output (include a listing of input tables and output result(s))
    Submit the project with tables S, CUST, PRODUCT and ORDERS (plus any other files or tables) as a test file for the program.
    Listing (documentation) of how to use the queries with example and display of the result for (a1, a2, ..., a4), (b1and b2) and (c1, c2, ..., c6)
i10. Any other supporting material for the project and/or team work

II. Blackboard

A designated team member should submit a copy of the final documentation listed in section I to the Osprey group account that has been named as **Team An information - Final Submission of Project**.

The documentation submitted to Osprey could be as a one piece word document or a window's zip file. Each section should be organized in the sequence of the sections in the I part and labeled with the sequence of i1, i2, i3, i4, i5, i6, i8, i9 (a1, a2, ..., a4, b1, b2, c1, c2, ..., and c6) and i10. For section i8, a complete program listing needs to be zipped as an independent section with the label 8i. In the window's zip file for each section of 8i and 9i, directories can be used and zipped.

III. Program Modules

As described earlier, the program should be designed in modular form, such that one module can take advantage of another. The team needs to use Java or C for the interface and implementation for a stand alone design. The preferred and recommended system for software development is Unix (Osprey). The program modules (components) should be submitted as a shar file using turnin on the UNF Osprey system.

Osprey

Each team will need to designate one of the members to submit a copy of the final software and related files on the Osprey system to the **abbassi.cop4710.sqlxmleng** turnin account as a shar file. The due date/time for turnin is firm on Monday, April 17 at 1:30 pm. No late assignment will be accepted after the due date/time.

The unshar file in the turnin should have all the necessary files for compilation (interpretation) and execution of the team program. The team should also provide sample files and queries and how to run the program on Osprey.

IV. Please make a copy of the submitted folder for your records.

The folder and its contents will not be returned.

Presentation Requirements

The projects will be presented on Monday, April 17 and Wednesday, April 19. Each team will have 20 minutes for presentation and questions/answers. Allow 15 minutes for the presentation and 5 minutes for the questions/answers. Each team should determine the member participation to best professionally present the team's SQL engine ideas. The instructor and members of the class will evaluate each team's design and implementation of the TSQLx engine.

A computer and projector will be available for the presentation. The team needs to be sure the classroom computer will work correctly during the presentation. The classroom machine may be checked on Monday, April 10.

Catalog Structure

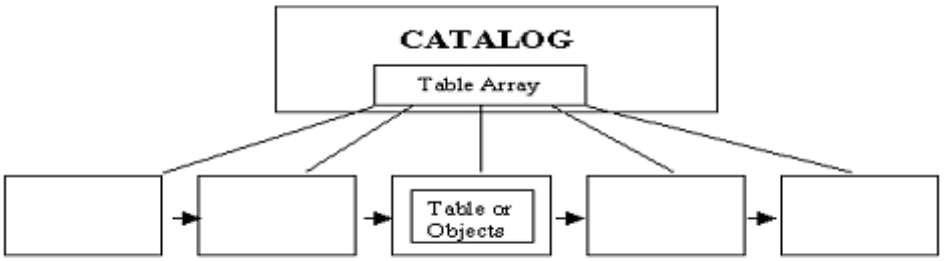

Table Structure