

CSCI235/MCS9235 Databases

Assignment 4 (Total 8 marks)

Scope

The tasks of this assignment include implementation and testing of XPath, XQuery for XML database, Transaction processing, NoSQL and normal forms.

This assignment consists of 6 tasks.

It is very important that you do what is written in **Prologue** section and you follow all recommendations written at the beginning of each task.

The deliverables from each task are listed at the end of specifications before the submission section.

A submission procedure is described at the end of assignment specifications.

Prologue

Download and unzip a file **all-files.zip**. You should obtain the following files:

bookshop.xml, a4create.sql, a4drop.sql, xlist.sql, xload.sql, xcreate.sql, xpath.sql, fxpath.sql, xdrop.sql, fxquery.sql.

Tasks

Task 1: XML schema (1 mark)

It is recommended to do Experiment 8.4 and 8.5 included in HOMEWORK 8.

Familiarize with the contents of XML document bookshop.xml. The document contains information about the books, journals and music CDs.

Create XML schema in a file bookshop.xsd and validate the XML document bookshop.xml against the XML schema.

Your XML schema must be implemented using a technique of typed element declarations. No other technique is acceptable!

Task 2: Implementation of queries in XPath language (1.5 marks)

It is recommended to do Experiment 9.1 included in HOMEWORK 9.

Familiarize with the contents of XML document bookshop.xml. The document contains information about the books, journals and music CDs.

The document bookshop.xml should be loaded to a database before you start this task. To access the document you can use SQL script fxpath.sql in sqlplus.

Implement the following queries in XPath query language and save each query in a file. The files are path1.xp, path2.xp, path3.xp, path4.xp and path5.xp accordingly.

(1) Find the title and authors of all books published before 2013.

(2) Find the titles, publishers and descriptions of books that have no keyword.

- (3) Find the title, volume number and issue number of a journal with ID value "263.18.37".
- (4) Find the titles of books or journals which price currency is "AU\$".
- (5) Find the titles and years of music CDs which category is "Classic".

Note: If you want to use UNIX ORACLE for XML database, you should download the scripts from Moodle site **Home works-> Scripts for XML database on UNIX ORACLE server.**

<http://www.xpathtester.com/xquery>

Task 3: Implementation of queries in XQuery language (2 marks)

It is recommended to do Experiment 9.2 included in HOMEWORK 9.

Familiarize with the contents of XML document `bookshop.xml`. The document contains information about the books, journals and music CDs.

The document `bookshop.xml` should be loaded to a database before you start this task. To access the document you can use SQL script `fxquery.sql` in `sqlplus`. The script accepts one inline parameter. A parameter passes a name of text file that contains a query in XQuery query language. The script reads a query and executes it against XML database of bookshop. For example, if your XQuery query is included in a file `query1.xq`, then to execute a query you have to connect to your Oracle account and you have to execute a script `fxquery.sql` at `SQL>` prompt in the following way.

`@fxquery query1.xq`

The results of query processing will be listed on a screen.

Implement the following queries in XQuery query language and save each query in a file. The files are `query1.xq`, `query2.xq`, `query3.xq`, `query4.xq` and `query5.xq` accordingly.

- (1) Find the titles and publishers of books that have at least three keywords.
- (2) Find each journal's title with the list of volume and issue pairs.
- (3) Find books or journals' titles which have comments.
- (4) Find how many books, or journals, or music CDs for each product. Display products' IDs and total number of items (books, journals and music CDs) for each product.
- (5) Find titles and authors of books which have been written by two or more authors.

Note: If you want to use UNIX ORACLE for XML database, you should download the scripts from Moodle site **Home works-> Scripts for XML database on UNIX ORACLE server.**

Task 4: Transaction processing (1 mark)

You **must** do Homework 10 before implementation of task 4.

Connect to your account on one of the Oracle servers (data-pc01 .. data-pc40) and execute a script `a4create.sql`. The script creates the relational tables of a sample database used in Assignment 4. Read the script and discover and draw a conceptual schema of the sample database.

You can use a script `a4drop.sql` to drop all relational tables created by `a4create.sql` later. Do not drop the relational tables now.

The files **`tread.sql`**, **`twrite.sql`**, and **`twrite2.sql`** contain parameterised SQL scripts that process a sample database. Consider each script as a separate database transaction and assume that the users can concurrently run the scripts/transactions. Assume that the script/transactions can run only at the following isolation levels: READ ONLY, READ COMMITTED, and SERIALIZABLE where READ ONLY is the lowest level and SERIALIZABLE is the highest level.

Decide the lowest isolation level for each script/transaction so that any possible concurrent execution of all scripts will never corrupt a sample database. Justify your answer in a file `task4.pdf`.

Justification of your answer must show that if a transaction runs at isolation level lower than the one selected by you then its concurrent execution with the other transactions may corrupt a database.

We expect that you provide the examples of concurrent execution that corrupt a database if a transaction runs at isolation level lower than selected by you. Such examples must be provided for all scripts/transactions considered in this task unless an isolation level selected for a script/transaction is the lowest possible. The examples must be presented in a "two-column" style used in Homework and lecture slides.

A correct guess without the comprehensive justifications scores no marks!

Task 5: NoSQL (1 mark)

You **must** do Homework 11 before implementation of task 5.

Consider the tables created in the script file `a4create.sql`. You will use Oracle NoSQL Database KVlite to implement a sample Key/Value database that contains information about Applicant, SPossessed, Position, SNeeded and Applies. The values of all attributes can be obtained from the script file `a4create.sql`.

Implement an application `LoadData.java` that inserts sample data into the Key/Value Database for the above tables. Assume that you have to store information about at least two applicants, at least two skills possessed by each applicant, at least two positions, at least two skills needed by each position, and at least two applies.

Implement an application `MatchSkills.java` that finds applicants that possess skills that match skills needed by a given position. That means the applicants possesses equal or higher skills than the skills needed by that position. **Note:** The applicants do not have to apply the position.

Task 6: Database normalization (1.5 marks)

Read and analyse the relational schemas and data dependencies valid for each one of the relational schemas given below and save your solutions in a file `task6.pdf`.

For each one of the relational schemas, determine the highest normal form, which is valid for a schema. Justify your answer. Justification must include the derivations of minimal keys from the functional dependencies and testing the validity of all normal forms (2NF, 3NF, BCNF) against the relational schemas, minimal keys, and functional dependencies.

A correct guess without the comprehensive justifications scores no marks!

(1) SUPPLIER(snumber, sname, city, postcode)

The attributes in a relational table SUPPLIER satisfy the following data dependencies:

- Each supplier identified by an attribute snumber has one name, lives in one city at one postcode.
- Each city has exactly one postcode.

(2) BRCUST(branch, customer, employee)

The attributes in a relational table BRCUST satisfy the following data dependencies:

- Each employee work at only one branch.
- Each customer contacts at a branch with only one employee.

(3) POWNER(pnumber, address, rent, onumber, oname)

The attributes in a relational table POWNER satisfy the following data dependencies:

pnumber \rightarrow address,
pnumber \rightarrow rent,
pnumber \rightarrow onumber,
onumber \rightarrow oname,

(4) UNIVERSITY(bldg#, campus, school, lecturer)

The attributes in a relational table UNIVERSITY satisfy the following data dependencies:

- Each lecturer belongs to only one school.
- Each school is located in one building.
- A building may host a number of different departments.
- A building has unique numbers at a campus.

(5) R = (T, N, O, C, E)

The attributes in a relational table R table satisfy the following data dependencies:

T \rightarrow N,
T \rightarrow E,
O \rightarrow C,
O \rightarrow T.

Deliverables

Task 1

Submit a file **bookshop.xsd** which contains the XML schema of bookshop.xml.

Task 2

Submit files **path1.xp**, **path2.xp**, **path3.xp**, **path4.xp** and **path5.xp** which contain the XPath scripts for the questions.

Task 3

Submit the files **query1.xq**, **query2.xq**, **query3.xq**, **query4.xq** and **query5.xq** which contain the XQuery scripts for the questions.

Task 4

Submit a file **task4.pdf** which contains the concurrent transactions of lowest isolation level with justifications.

Task 5

Submit a file **LoadData.java** and a file **MatchSkills.java**.

Task 6

Submit a file **task6.pdf** with a solution of the problems in Task 6.

Submission procedure

Zip the files bookshop.xsd, path1.xp, path2.xp, path3.xp, path4.xp, path5.xp, query1.xq, query2.xq, query3.xq, query4.xq, query5.xq, task4.pdf, LoadData.java, MatchSkills.java and task6.pdf into a file assignment4.zip.

- (1) Connect to Moodle.
- (2) Navigate to a folder ASSIGNMENT SUBMISSIONS
- (3) Click at Assignment 4, Submit your solution here link.
- (4) Click at Add Attachments button.
- (5) Navigate to a location where a file **assignment4.zip** has been saved.
- (6) Select the file and click at Open button.
- (7) Click at Submit button.
- (8) Click at OK button to return to Home Page.

A policy regarding late submissions is included in the course outline.

The assignment must be submitted as **soft copy** only.

The assignment is an **individual assignment** and it is expected that all its tasks will be solved **individually without any cooperation** with the other students. If you have any doubts, questions, etc. please consult your lecturer or tutor during lab classes or office hours. Plagiarism will result in a **FAIL** grade being recorded for that assessment task.

Late submissions do not have to be requested. Late submissions will be allowed for a few days after close of scheduled submission (up to 3 days). Late submissions attract a mark penalty; this penalty may be waived if an appropriate request for academic consideration

(for medical or similar problem) is made via the university SOLS system *before* the Due date. No work can be submitted after the late submission time.

Marks and comments on the assignments will be returned to the students in two weeks after the submission on Moodle.

End of specification