WESTMINSTER
INTERNATIONAL UNIVERSITY IN TASHKENT
An Accredited Institution of the University of Westminster (UK)

Module name and code	Database Systems Development, 5BUIS009C
CW weighting	40%
CW number and title	CW2 Refer/Defer
Lecturer setting the task with contact details and office hours	Dmitriy Pochitaev dpochitaev@wiut.uz Monday 11:00-12:00
Submission deadline	TBD
Results date and type of feedback	Written feedback
The CW checks the following learning outcomes:	
1. Optimise queries in achieving maximum execution performance; 2. Report and formulate data according to users specification; 3. Attach a database to the Internet for retrieving and storing data through a web interface; 4. Analyse and implement issues related to object relational databases;	

Task

This is an individual coursework. CW2 is a logical continuation of CW1. You need to develop a web interface for the DB system. It is recommended to use the same case study as you have used for CW1, but you also can choose another case study.

Task description

1. **Report [weight 15%]**
   - Relational DB diagram, e.g. using diagramming tool of SQL Server Management studio.
   - Evaluate and reflect on the system you have built, what problems you had during system implementation and how you solved them.
   - Black box testing of major functionalities. For each test provide test description, expected results, actual results and comments on your test results. Sample table for test results presentation see in “General requirements” section.

2. **Practical part [weight 85%]**
   For implementation of the web application you should use ASP.NET MVC framework, for database connectivity should be used ADO.NET, Entity Framework or Dapper. You should use MS SQL Server *LocalDB* as a DBMS for your application.

Implementation of the prototype

Web application functionality

Depending on your case "item" could be a product (e.g. furniture, property, flowers etc.) or some kind of service.

1) **Item CRUD (Create Read Update Delete) using SQL Server stored procedures/functions.** Item table should contain at least 8 fields including at 1 image (or other binary data field), 1 date/time field, 1 Boolean field.

2) **Report (with paging, sorting and filtration)**
System should allow to display data relevant for your case. Data should be taken from **at least 4 tables**. Filter form should contain 3 input fields (e.g., if item is a flat to rent search could be done by number of rooms, floor, address etc.).
   - Search results should support sorting in ascending and descending order. It should be possible to sort by at least 3 columns.
   - Search results should support paging.

   At the database level you should use stored procedure or function for selection of data.

3) **Data export**
It should be possible to export search results from the point 3) to XML or JSON. Generation of XML or JSON should be done using stored procedure or function.

4) **Data import**
Implement a web form that will allow importing of items from the file. Import should be possible from XML or JSON formatted files. Provide sample XML or JSON file for testing import. Import should be done using stored procedure.

5) **Query performance optimization**
Analyse your SQL select statement in point 2) using “explain” SQL command (in SQL Server Management Studio use Ctrl + M shortcut). Suggest indexes that will improve SQL query performance. Explain your choice of indexes, support your answer with the output of the SQL “explain” command. Include these explanations in a form of SQL multiline comments (/* ... */) in the SQL script file that you will submit.

Required triggers

- ➢ A Trigger that will disallow insert of items on Saturdays and Sundays.
- ➢ A Trigger that will perform validation of some business rule (constraint).

General requirements

a) Database connection string should be specified in a *configuration* file. If this requirement is not implemented you will be penalised by 10 marks.

b) Use SQL Server LocalDB (database files <db name>.mdf and <db name>.ldf), this way the complete database will be shipped with your project.

c) Database must be well populated with sample data. Your *LocalDB* SQL Server database must contain enough data in order to test your web application (at least 10 records per table). If this requirement is not implemented you will be penalised by 10 marks.

d) Your web application should provide clear navigation scheme, all sections must have a link in a navigation menu, looks accurate, and follows general usability principles. You could use ready HTML templates.

Testing

Perform black box testing of all the functionalities of your application (points 1-4 from Practical part and triggers) and present results of your tests. Overall expected at least 7 test cases. To present your tests and test results you must use the template provided below:

Test description	***Expected results***	***Actual results***	***Comments***
			If expected and actual results do not match, provide explanations here.

Submission details

Coursework *must be submitted electronically through the Intranet system*. Zip all files and name it according the following template:

DBSD.CW2.IDnumber.zip

Example: **DBSD.CW2.5678.zip**

Submit the zip file through the Intranet submission system.

Zip file should contain:

a) Report

b) Text file with SQL for creation of triggers, stored procedures, and functions. Also it should include indexes and explanations of created indexes as specified in point 6 of the practical part)

c) XML or JSON files for data import

d) Source code of your web application - all ASP.NET files including VS.NET project and solution files.

e) SQL Server LocalDB files

Sample structure of the zip archive:

```
DBSD.CW2.5678.zip
                |- DBSD.CW2.5678 _Report.docx
                |- Triggers.sql
                |- Procedures.sql
                |- Indexes.sql
                |- Import.xml
                |- Import.json
                |- DBSD.CW2.5678.csproj
                |- DBSD.CW2.5678.sln
                |-AppData\
                   |-(e)<your DB>.mdf
                   |-(e)<your DB>.ldf
                .... other source code files in your web application
```

Important!

It is your responsibility to ensure that all submitted files are readable and not corrupted. Please double check this before submitting your files.

Note: you might be asked for a viva if any questions arise concerning your application.

Appendix A Case study selection instructions

You should decide on type of system to be modelled and then provide a short case study description. You are free to choose a type of system you feel most comfortable with. Please not that you will need at least 5 tables in your database.

In the report you should provide description of selected case study. If you have any questions/difficulties with selection of the case study, please contact your tutor for advice.

Marks allocation

Task	**Mark**
Report	**15%**
DB diagram	3
Self-evaluation	2
Black box testing, all major functionalities should be tested, test results presented in clear and well-structured format, tested various scenarios (e.g. correct data entered, incorrect data entered)	10
Practical part	**85%**
Item CRUD (Create Read Update Delete) using SQL Server stored procedures/functions	18
Report (with paging, sorting and filtration)	16
Data import	14
Data export	14
Query performance optimization	6
HTML design (navigation, overall usability, design template was used)	6
A Trigger that will disallow insert	6
A trigger for business rules/constraints	5
Database must be well populated with sample data and use SQL Server LocalDB files– if not deduct up to 10 marks	*-10*
Database connection string should be specified in a *configuration file – if not deduct up to 10 marks*	*-10*