

Assignment 1

Posted Date: Feb 2, 2023

Submission Due: Feb 23, 2023 (11:59 pm)

Late assignments will not be accepted and will result in a 0 on the assignment**Objective:** This assignment covers two learning objectives (lo).

- **lo#1:** Solve a data management problem, by building a database in a systematic manner. Understand the problem, build data model, and map the model to a physical design – To achieve this task, you need to perform conceptual design, logical design, and physical design in a sequence.
- **lo#2:** Build a prototype of a light-weight DBMS using Java programming language, which performs at least 2 required functions, and 1 optional function from a given list of DBMS functions. If your application performs more than 1 optional functions of DBMS then you may get additional points for novelty. [Note: Your code must be written by you, and will be checked for academic integrity]

Plagiarism Policy:

- This assignment is an individual task. Collaboration of any type amounts to a violation of the academic integrity policy and will be reported to the AIO.
- Content cannot be copied verbatim from any source(s). Please understand the concept and write in your own words. In addition, cite the actual source. Failing to do so will be considered as plagiarism and/or cheating.
- The Dalhousie Academic Integrity policy applies to all material submitted as part of this course. Please understand the policy, which is available at:
https://www.dal.ca/dept/university_secretariat/academic-integrity.html

Assignment Rubric - based on the discussion board rubric (McKinney, 2018)

	Excellent (25%)	Proficient (15%)	Marginal (5%)	Unacceptable (0%)
Completeness including Citation	All required tasks are completed	Submission highlights tasks completion. However, missed some tasks in between, which created a disconnection	Some tasks are completed, which are disjoint in nature.	Incorrect and irrelevant
Correctness	All parts of the given tasks are correct	Most of the given tasks are correct However, some portions need minor modifications	Most of the given tasks are incorrect. The submission requires major modifications.	Incorrect and unacceptable
Novelty	The submission contains novel contribution in key segments, which is a clear indication of application knowledge	The submission lacks novel contributions. There are some evidences of novelty, however, it is not significant	The submission does not contain novel contributions. However, there is an evidence of some effort	There is no novelty
Clarity	The written or graphical materials, and developed applications provide a clear picture of the concept, and highlights the clarity	The written or graphical materials, and developed applications do not show clear picture of the concept. There is room for improvement	The written or graphical materials, and developed applications fail to prove the clarity. Background knowledge is needed	Failed to prove the clarity. Need proper background knowledge to perform the tasks

Citation: McKinney, B. (2018). The impact of program-wide discussion board grading rubrics on students' and faculty satisfaction. *Online Learning*, 22(2), 289-299.

Explanation of the rubric: Suppose you received different grades in Clarity for the 2 problems

Problem #1: 25% in clarity

Problem #2: 15% in clarity

Then your overall grade for the clarity will be avg of (25+15) % = 20%

Problem 1: Data Modelling related problem. Objective is lo#1

Follow all the steps in the given order. For you, some steps could be optional and depends on the solution you create.

Step 1. Visit <https://www.dal.ca/> and sub-pages within dal.ca domain. **Note:** You do not have to consider dalonline.ca.

Step 2. Identify at least 9 unique entities (sub-types are not counted as separate entities for this problem; Only the supertype will be considered as 1 entity), which must contain at least 1 weak entity. Write your assumption in a short paragraph explaining why you considered these entities, and how do you justify your strong entities as strong and weak entities as weak.

Step 3. Now using Chen model create an ERD or EERD. Your ERD/EERD should highlight all the possible attributes for the entities. In addition, highlight what types of attributes are these, e.g. primary key, partial key, multivalued attribute etc. **Note:** Your ERD should contain cardinality information, and it should be built using a standard data modelling tool, such as Erwin, Visio, draw.io etc.

Step 4. Identify if there are any design issues in your ERD. If you find any, then write a short paragraph about the issues, and fix it by generating a refined ERD/EERD

By the end of Step 4, you will be completing a **conceptual model**. Now, it is time to move to **Logical mapping**.

Step 5. Consider each entity from your data model as a table. Now, create a tabular structure with the attributes only (no need to consider any data points; just attributes are needed), and write a short paragraph and/or couple of bullet points to highlight the dependencies that exist in your **logical model**

Step 6. If required, perform some level of normalization (1NF->2NF->3NF), and write your assumptions on why you considered this normalization.

Step 7. Now, create a database, and empty tables by writing DDL SQL query in MySQL Workbench. This will give you the **physical model**.

Problem 2: Prototype of a light-weight DBMS using Java programming language (no 3rd party libraries allowed). Objective is lo#2

Follow the given requirements strictly.

Req #	Task	Checklist
1	Use a standard Java IDE to develop your application. Any JDK version is acceptable	
2	While writing the Java code, follow JavaDocs specification for commenting styles, such as @param, @return etc.	
3	Write/Draw (using any tool), the design principles that you will be using or have used in your application program development/execution. Check SOLID design online	
4	Your application should be console based (no GUI needed) and it should accept user input in the form of SQL query, once the user has successfully logged in. You should provide functionality for creating one database only	
5 Required	One of the required functionalities of your application is creating two factor authentication - "user authentication module". It should use ID, Password, and question/answer for authentication. <ul style="list-style-type: none"> You may wish to design a class to handle this authentication. The authentication is needed because the application should support multiple users. For any hashing you can use standard Java library, such as md5 	
6 Required	The second required functionality is design of persistent storage. Once the input query is processed -- your data, user information, logs etc. must be kept in a file format, which is custom designed . You cannot use JSON, XML, CSV etc. You need to design your own delimiters to store or access the data within a text file.	

7	Your database management system should support one additional function from the given set of functions, which are numbered as 7(1) to 7(3). You can implement any one of the following functions, but if you implement more, you will get novelty points	
7(1)	<u>Implementation of Queries (DDL & DML)</u> – CREATE, SELECT, INSERT, UPDATE, DELETE applied to any number of tables. Hint: You can create a class called query, and for every query, you can build a separate method	
7(2)	<u>Implementation of Transaction</u> – Implement a single transaction handling logic, where a transaction is identified by the system depending on user input, such as “Begin Transaction”, “End Transaction” etc. Hint: Since this is a transaction, it must follow ACID property, therefore, the processed query should not immediately write on your custom-made database text file. You can process the queries but keep those in intermediate data structure, like LinkedLists, ArrayLists etc. Once it gets “Commit” only then it should update/delete/insert etc. in the text file from the data structure. For “Rollback” the data structure must be emptied	
7(3)	Implementation of ERD – Implement an ERD generation class, which can scan through your data, structure etc. files and generate a console-based ERD. It should be created based on the current state of the database, and also add cardinalities. The primary key, and foreign key can be shown as Pk, and Fk. Hint: For the ERD, you can either print the structure and relationships on screen as textual output or you can use for loop with “ ”, “-” to print rectangle boxes for entity representations and relationships.	

Submission Guidelines:

1. All written reports, images, code etc. must be added in a folder, and compress it with **.ZIP** format only.
2. If not mentioned by TAs, then please rename the .zip file with your **B00xxxxx_FnameLname_A1**
3. Submit your Java code in gitlab. Your TA must have provided guidelines for that. If not, please ask the TA.
4. You must include Test Cases (at least 3) for the developed application and provide necessary screenshots as evidence of testing.
5. Check the next point “Suggestions” for quality improvement and time management.

Suggestions:

Better Quality: To obtain good grades, you should follow the points given below:

- Try to understand the assignment requirement and follow all the steps required.
- Do not miss adding citations. If you write a single sentence taking the idea from somewhere else, then give credit to the author. Therefore, provide citation for any report you write, or any code you implement
- When you add citation, make sure to add it in a standard format and uniform format. E.g. if I refer 3 sources for writing a report, then I must cite the 3 sources in same format. One source in MLA, two sources in APA citation format will be a mismatch. Therefore, follow any one standard citation format
- Make sure to provide inline citations within report, and programming code
- Any image/picture/flowchart/diagram you add, make sure to provide a caption and a number for that image. It should be placed at the bottom of the image. E.g. “**Fig 1: Weekly time management chart for CSCI 5408**”
- Any table you add, must have a number and caption. This should be added on top of the table. E.g. “**Tab1: Table highlights the requirements in a ordered format**”

Time Management: Follow proper time management to reduce stress, and last-minute preparations. I am suggesting you follow the pie chart, which will require you to spend 5 hours in a week outside the classroom time for this course.

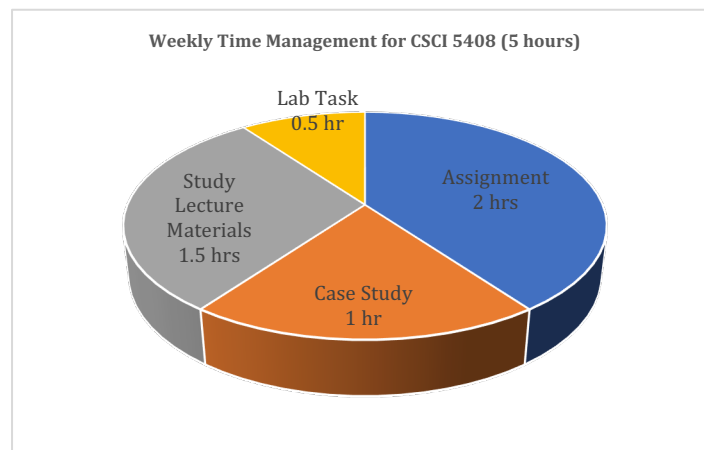


Fig 1: Weekly time management chart for CSCI 5408