Assignment #2

CSCI 5408 (Data Management, Warehousing, Analytics)
Faculty of Computer Science, Dalhousie University

Date Given: Jun 4, 2022
Due Date: Jun 18, 2022 at 11:59 pm

Late Submissions are not accepted and will result in a late penalty of 10% deductions / day in the assignment.

Disclaimer: This assignment requires students to work on research paper, open Datasets, and RDBMS with appropriate citation. Submissions related to this assignment will not be used for commercial purposes.

Objective:

- The objective of this assignment is to understand research and industry problems related to distributed database operations, and transactions management.

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

	Excellent (25%)	Proficient (15%)	Marginal (5%)	Unacceptable (0%)	This Rubric Applied to
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	Problem #1 Problem #2
Correctness	All parts of the given tasks are correct	Most of the given tasks are correct However, some portions need	Most of the given tasks are incorrect. The submission	Incorrect and unacceptable	Problem #1 Problem #2

		minor modifications	requires major modifications.		
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	Problem #1 Problem #2
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	Problem #1 Problem #2

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

Problem #1: This problem contains a programming task

Research & Development of a Distributed Database

You need to work on the dataset or the link (https://parks.novascotia.ca/) that you have used in your assignment #1 (problem #2). You need to do the following:

1. Convert your final ERD/EERD to a physical design and each entity can translate into one table in your physical design. **Note**: There is no minimum or maximum number of records (data points) set for each table. Depending on data availability, you can add 4+ data rows for each table.

Local Database

2. Create a single database (name of the database "*A2_loc<your_netid>*") in your **local MySQL RDBMS**, and add all the tables within this database.
    a. Write one transaction block with 1 **SELECT** and 1 **UPDATE** operation (your choice)
    b. Execute the transaction and record the execution time (in milliseconds)
    To get time in milliseconds, you can use **mysql>set profiling = 1;**
    To show time in milliseconds, you can use **mysql>show profiles;**
    c. Try adding one more SELECT and one more UPDATE operation, and measure the execution time
    d. Record the 2 transactions and the 2 execution times in a tabular format.

Remote Database

3. Create a single database (name of the database "*A2_rem<your_netid>*") in your **remote MySQL RDBMS** (GCP), and add all the tables within this database.
    a. Write one transaction block with 1 **SELECT** and 1 **UPDATE** operation (same as point number 2)
    b. Execute the transaction and record the execution time (in milliseconds)
    To get time in milliseconds, you can use **mysql>set profiling = 1;**
    To show time in milliseconds, you can use **mysql>show profiles;**
    c. Measure the execution time with the updated transaction (same as point number 2 - c)

d. Record the 2 transactions and the 2 execution times in a tabular format.

Distributed Database

4. Create two databases with identical names (name of the database "*A2_dist<your_netid>*"), one in your **remote MySQL RDBMS** (GCP), and another in your **local MySQL RDBMS.** Distribute the tables in both locations equally (if possible). E.g. if there are 5 entities in the original model, then after table creation, you can keep 3 tables in remote location, and 2 tables in the local RDBMS.
   a. Write a Java code to connect the two databases (same as the lab task)
   b. Write one transaction block with 2 **SELECT** and 2 **UPDATE** operation
   c. You need to ensure that within the same transaction, you perform 1 SELECT and 1 UPDATE operations in local site, and 1 SELECT and 1 UPDATE operations in remote site. This will make the transaction, a distributed transaction.
   d. Execute the transaction and record the execution time (in milliseconds)
      To get time in milliseconds, you can use **mysql>set profiling = 1;**
      To show time in milliseconds, you can use **mysql>show profiles;**
   e. Record the distributed transaction statements and the execution time in a tabular format.

Problem #1 Submission Requirements:

- Approx. 1.5-page Report (***problem1.pdf***) containing the summary of observation, explanations, and analysis
- SQL dump of only local, only remote, and distributed database containing table structure + value
- You can include the screenshots of the output in ***problem1.pdf***
- Java code for the distributed database, and the content of GDC/GDD showing the distribution of tables for point number 4 should be in gitlab.

Problem #2: This problem contains a programming task

Product Development: Using Core Java code, you need to create a Lock manager for RDBMS. You do need to use any packages other than the core Java concept. Use JDBC to connect to MySQL

1. Consider the Ocean tracking databased that you created in Assignment 1. If you did not work on the problem in assignment 1, then ask your TA to provide one SQL dump for that database.
2. Write a Java program to perform two concurrent transactions.
   a. Each transaction containing 5 statements, such as UPDATE, SELECT, DELETE etc).
   b. Both transactions should perform the operation on the same field. This is important to examine how consistency is maintained.
   c. In your Java code prepare all the SQL queries for both transactions as two separate blocks. Do not execute individual queries
   d. Set autocommit to false in your Java code
   e. Now, write 3 methods and call these methods in the following order from the main program. This is the implementation of 2-phase locking protocol.

acquireLock(MyTransaction t): This method will be called to acquire locks by one transaction
operation(): This method will be called to perform the actual set of operations after acquiring the locks. It means the prepared SQL statements for each transaction that acquired the locks will be executed in this method.
releaseLock(): This method will be called once operation of a transaction ends

Problem #2 Submission Requirements:

1. Upload your program code to gitlab (https://git.cs.dal.ca). Follow naming convention given by TA. Or ask your TA about that.
2. Provide Queries of all transactions you performed
3. Provide screenshots of all possible outputs and test cases.