MACQUARIE
University

FACULTY OF SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

COMP2350/COMP6350:

Database Systems

Session 2, 2023

Assignment Three: Procedural Programming

DUE DATE: 11:55 pm Sunday 29th October (Week 12)
Total Marks: 100 (10% of the final grade)
Objectives: ULO1, ULO2, ULO3 and ULO4 (refer to Unit Guide)

This is an individual assignment. Part A has 4 database programming and implementation tasks. Part B has a report writing task that presents the implementation and test of Part A tasks. Marks are allocated to Part A tasks; however, if a Part A task is not documented in the report (Part B task), 0 (zero) mark will be awarded to that Part A task.

General requirements for all tasks *(when applicable):*

1. Complete the table below and include it at the beginning of each required document listed on page 3 in the Submission section:

Unit Code		Assignment#	3
Student ID Number		Student Name	
Tutor's Name		Workshop Date/Time	

2. Font Size must be 11 or 12 points.
3. Line Spacing must be single or 1.5 lines
4. Digitalized hand-written or hand-drawn contents will not be accepted for assessment.
5. ChatGPT is forbidden in this unit.
6. Late submission penalty: see COMP2350/COMP6350 Unit Guide for 2023 S2.

Assignment Background

City-145 library is planning to automate their manual process loan system. As a COMP2350/COMP6350 student, you are tasked to contribute to their database implementation (Part A) using MySQL and present a report (Part B) to the Planning Team. Business rules that are relevant to your tasks are provided below.

Business Rules:

BR1. Any book can be borrowed if and only if at least one copy of the book is being holding at the branch.

BR2. Only members with "REGULAR" member status can borrow books.

BR3. Each member can borrow one copy of the same book on the same day.

BR4. A member can borrow up to 5 items for 3 weeks (i.e., 21 days).

BR5. The return due date of the borrowed book cannot be past the membership expiry date.

BR6. If a member has an outstanding fee* and it has reached $30, his/her membership will be suspended.

BR7. If a member has an overdue item, his/her fine fee will be increased $2/day passing the expiration date and the membership will be suspended.

BR8. When a suspended member clears their fine (i.e, paid all the outstanding fees) and has no or has returned all overdue items, reset the member's membership status to "REGULAR".

*outstanding fee can include fee for lost, damaged or overdue items, late membership fee.

Part A: Database Programming and Implementation

Use what has been covered during week 5 – week 11 (both lecture and workshop, including relevant textbook chapters) on SQL and Procedural Programming to complete Tasks 1 – 4. If you use anything that was not covered during week 5 – week 11, your implementation for that task will not be marked.

TASK 1: (10 marks)

Use the provided .sql file to create and populate required tables.

> **Note:** For all tasks of this assignment, do not delete or modify any provided data (in the .sql file). If more data is required to complete the task, insert appropriate data into relevant table(s) when need.

- (5 marks) Update the Member table to allow logging fine fees for overdue.
- (5 marks) Verify the correctness of the above operations.

TASK 2: (30 marks)

(15 marks) Write a trigger to implement the BR8 business rule listed on page 2.
(15 marks) Exceptions must be handled by error handlers.

TASK 3: (30 marks)

(15 marks) Write a stored procedure to list the members that currently have an overdue item and their (individual) membership has been suspended twice in the past three years. End these members' membership by setting their MemberStatus to "TERMINATED".

(15 marks) Error handler must be implemented to handle exceptions.

TASK 4: (30 marks)

(15 marks) **Task2-Test:** Think about what data you would need to test different data scenarios. Outline those scenarios in a "test plan" section of your submission to help you verify the correctness and completeness of your trigger you wrote for Task 2. Add necessary data into relevant table(s) for testing all possible cases that you can think of. If the trigger is not working, marks will not be awarded for Task2-Test.

(15 marks) **Task3-Test:** Think about what data you would need to test different data scenarios. Outline those scenarios in a "test plan" section of your submission to help you verify the correctness and completeness of your stored procedure you wrote for Task 3. Add necessary data into relevant table(s) for testing all possible cases that you can think of. If the stored procedure is not working, marks will not be awarded for Task3-Test.

Part B: Report Writing

- Present the schema for the tables you created. Clearly indicate relevant keys (primary, foreign, unique, …).
- Describe how you have completed each of the 4 tasks in Part A. Include the screenshots as evidence and brief justification.

Submission:

- A **.sql** file: Submit a sql text file consisting all programming and implementation for all 4 tasks of Part A.
- A **.pdf** file: Submit a pdf document for Part B.