Student ID: ____

Full Names: ____

Midterm Exam – Part 2_Q1

(April 2025)

Author: Professor Obinna Kalu, M.Sc., MSCS

1. The time allotted for completing the entire exam (both Parts 1 and 2) is $2^1/_2$ hours.
2. **This Exam belongs to MIU CS Department and must not be taken away, or copied or photographed or reproduced or transferred or shared or distributed**. **Any violation will be penalized.**

Make sure to include the screenshots of your work/results, where required.

Midterm Exam – Part 2_Q1 (90 points)

Part 2_Q1: Software Requirements, Analysis, Design and Implementation (90 points)

1. (90 points) Software Requirements Analysis, Data modeling, Data Design and Implementation

Consider the problem statement/description given below.

Your tasks for this question will include:

- Perform Analysis and Domain modeling to create a Domain model UML Class diagram for the Software solution.

- Perform Data modeling and create an E-R model for the database needed for the system. You may draw the E-R diagram using a graphical drawing tool on computer or draw by hand using paper and pen/pencil.

- Implement your model onto a physical Database

- Write and execute some Queries

Problem Statement/Description:

Assume that a state bank named, CS489 Bank, has hired you to design and develop software solution for their Retail Banking system, which they will be using to manage data about their Customers and the Accounts they own and operate.

For the purpose of this Exam, here is a simplified description of the data model for the system:

A **Customer** can have many **Accounts**.

And, an **Account** can be owned by one or many **Customers**. An Account that is owned by multiple Customers is known as a Joint Account.

An Account can be of one AccountType.

Each AccountType can have many Accounts.

An Account will have many Transactions done on it.

Each Transaction must be associated with just one and only one Account.

Note: There CANNOT be an Account without a Customer. And every Customer MUST have at least one Account.

IMPORTANT: Your solution model should include the following four entity classes, named:

1. Customer
2. Account
3. AccountType
4. Transaction

Here are the attributes for the **Customer** entity, including some useful descriptions and/or sample data values:

Customer:

customerId: Integer, (Primary Key field)

firstName, (required field)

lastName, (required field)

telephoneNumber (optional field)

Here are the attributes for the **Account, AccountType and Transaction** entities, including some useful descriptions and/or sample data values:

Account:

accountId: long, (Primary Key field)

accountNumber, (required field, unique)

dateOpened, (required field) (e.g. 2019-01-17, 2023-02-20, etc.)

status, (REQUIRED FIELD) (e.g. Active, Dormant, etc.)

balance (required field) (e.g. $100.00, $950.00 etc.)

AccountType:

accountTypeId: Integer, (Primary Key field)

accountTypeName, (required field)

Transaction:

transactionId: (Primary Key field)

transactionNumber: (Required field, Unique)

description, (required)

valueAmount: (required)

transactionDate: (required)

transactionTime: (required)

transactionType: (optional)

Data:

Here is the bank's data, which you are expected to input/load into your database:

Customers data: (Note: This is NOT necessarily the Database table)

CustomerId	First Name	Last Name	Telephone
1	Daniel	Agar	
2	Bernard	Shaw	(641) 472-1234
3	Carly	DeFiori	

Accounts data: (Note: This is NOT necessarily the Database table)

AccountId	Account Number	Date Opened	Status	Balance	CustomerId(s)	AccountType
1	CK1089	2021-10-15	Active	105,945.50	3	2
2	SV1104	2019-06-22	Active	197,750.00	1, 2	1
3	SV2307	2014-02-27	Dormant	842,000.75	3	1
4	LN4133	2005-11-07	Active	674,500.00	3	3

AccountTypes data: (Note: This is NOT necessarily the Database table)

AccountType Id	AccountType Name
1	Checking
2	Savings
3	Loan

Transactions data: (Note: This is NOT necessarily the Database table)

Transaction Number	Description	Value Amount	Transaction Date	Transaction Time	Transaction Type	Account Id
D0187-175	Deposit to Savings	100,000.00	2021-10-15	12:15:00	Deposit	2
W1736-142	Teller counter withdrawal	550.00	2022-08-24	10:05:00	Withdrawal	1
DD001-142	Direct deposit – wage	2475.75	2014-03-01	05:00:00	Direct deposit	1
P162-0017	Merch purchase online	150.95	2019-12-15	14:19:00	Purchase	1

For this question, you are required to do the following tasks:

TASK 1: Domain modeling

Draw the Domain model UML class diagram for the solution. Your diagram must show the four Domain entity Classes, the Attributes, Relationship(s) and Multiplicities etc.

TASK 2: Database Design and E-R modeling

Create an E-R model for the database needed for the system. You may draw the E-R diagram using a graphical drawing tool on computer or draw by hand using paper and pen/pencil.

TASK 3: Database implementation

Implement your E-R model on a physical Relational (or Non-relational) database. Use any RDBMS (or NoSQL DB) of your choice e.g. MySQL, MongoDB etc. Make sure to implement ALL the requirements as given in the specification above. E.g. Required fields, Unique fields etc.

TASK 4: Data

Populate the Database with all of the Bank's given data

TASK 5: Implement and execute Queries

Write and execute Queries for the following requirements:

- Display the list of ALL the Accounts registered in the banking system, sorted in descending order of the Account Balances. Include the Customer data for each Account.
- Display the list of ALL Transactions with a Value Amount greater than 500.00. Include in the result, the Account Numbers. And sort the list in ascending order of the Tansaction Date and Time.

Submissions for the above tasks:

Task 1: Save or export the UML diagram to a .PNG or .JPG image file, which you include in your zip file for submission/upload. Or hand-in your hand-drawn diagram.

Task 2: Save or export the E-R model diagram to a .PNG or .JPG image file, which you include in your zip file for submission/upload. Or hand-in your hand-drawn diagram.

Task 3: Save (or export) ALL the SQL DDL Code in a filenamed, myCS489BankDBScript.sql which you include in your zip file for submission/upload. Or you make take screenshots of your work.

Task 4: Save (or export) ALL the SQL DML Code in a filenamed, myCS489BankDBDataPopScript.sql which you include in your zip file for submission/upload. Or you make take screenshots of your work.

Task 5: For each of the Queries, execute your code and take a screenshot of your computer showing your code and the output result, and save/export it to a .PNG or .JPG image file, which you include in your submission. Or save the Query into a .sql file which you include in your zip file for submission/upload.

Enjoy!

//-- The End --//