



UNIVERSITY OF
LINCOLN

University of Lincoln Assessment Framework

Assessment Briefing Template 2021-2022

NOTE: All Assessment Briefings should be made available prior to the commencement of the module, clearly signposted on the module Blackboard site as well as included in any module handbook or briefing document.

Module Code & Title: CMP2806M Scalable Database Systems

Contribution to Final Module Mark: 100

Description of Assessment Task and Purpose:

A typical bank operation brief

One of the local banks has got in touch with your firm to help them build a system that allows better management of bank processes. As any bank, your client deals with opening, managing and closing customer *Bank Accounts*. To open a bank account, a bank customer is required to provide their full name, date of birth, address, and telephone number. They are also required to deposit an initial sum of money called an opening balance (opening balance > £50). Once this is provided the bank will open a bank account and provide their new customers with a sort code and a unique account number. The bank will then manage the customer *Transactions* (incoming transactions - money coming into the account, and outgoing transactions – money leaving the account). The bank is also offering *Loans* to some of the customers and therefore receiving loan payments from customers on a particular date every month. When a loan is given to a customer (only customers who have a bank account can have a loan) the bank stores the monthly payment rate (e.g. £300), the number of monthly payments (e.g. 25 months), the first payment date (e.g. 01.02.2020), and the monthly due date (payment leaves the account on the 25th of day of every month).

You are required to design and implement a MySQL database schema to be used to store relevant data for the bank in line with the basic operational scenario as explained above.

Requirements:

1. Create a database capturing the base level of entities and relationships as explained in the brief above. Please include all appropriate tables, assumptions, constraints, and relationships (e.g. explain things such as setting a primary key and reason provide reasons for your choice).
2. The database should be normalised, at least to 2NF.

3. Populate and add records to the tables. Assume you have at least 10 customer accounts with half of them having loans. Please make sure you to have a transaction history for at least 1 month for some of the customers.

4. A number of SQL queries should be defined to analyse customers, bank accounts, transactions and loans data. These queries are:

4.1. List all bank customers (including their name and account number) who have their loan payment due in the first 7 days of the month.

4.2. Extract all bank transactions that were made in the past 5 days (please include customer and account details).

4.3. List the customers with their current balance greater than £5000 (at least 1 customer should have their current balance greater than £5000). The current balance can be calculated by summing the opening balance of the account, all the incoming transactions of the account, and deducting the outgoing transactions of the account.

4.4. The Bank management team often needs to know the "Total Outstandings" of the bank. The Total Outstandings is the up-to-date balance over all bank accounts and includes the sum of the opening balance of every bank account deducting all outgoing payments and adding all incoming payments of every bank account. Write an SQL query to calculate and list the Total Outstandings of the bank.

5. Choose any two queries from the above and implement them as Procedures.

6. To document your work, write and submit a report that describes how your design choices and operational aspects of your database. The report should include:

6.1. an overview of the design process.

6.2. a schematic/diagram of the tables and their relationships.

6.3. a listing of the queries, i.e. the SQL syntax for each query, AND their outputs.

6.4. anything else seen as relevant.

You must implement your final database solution, via an SQL script, using the phpMyAdmin and MySQL database platform. In terms of deployment of the SQL script, your solution must only be implemented and deployed with the tools given to you through the delivery team.

Your SQL script for this task should include MySQL DDL code for creating your final database solution on your phpMyAdmin/MySQL, and should include ALL the following BEFORE carrying out other parts:

1.1. Creation of the database from scratch using the correct commands;

1.2 Creation of database tables and columns, completed with integrity constraints (i.e. primary keys, foreign keys, UNIQUE and NOT NULL).

Learning Outcomes Assessed:

LO1 Through development of key concepts of database theory, use appropriate tools and techniques to design a database.

LO2 Use cognition to appraise the structure of a database design using standard evaluation mechanisms.

LO3 Implement and use a relational database using industry standard query language.

Knowledge & Skills Assessed:

Ability to analyse and database requirement and build a good SQL database system for a given scenario.

Assessment Submission Instructions:

This assessment is an individually assessed component – all submitted assessment material is passed through the university Turnitin plagiarism detection tools. Your work must be presented according to the School of Computer Science guidelines, and where identified in this assessment briefing document, meet any further requirements. Please make sure you have a clear understanding of the grading principles for this component as detailed in the associated CRG.

Date for Return of Feedback:

The deadline for submission of this work is included in the hand in sheet on Blackboard.

Format for Assessment:

Database design and implementation.

Required submission documents:

1. A report. This should be in PDF format.
2. .sql script.

Every student should individually upload their .sql script to the assessment supporting documentation upload area (NOT Turnitin!); if you failed to do that a penalty will be applied.

Every student should individually upload their .pdf Report to the Assessment Turnitin upload area; failure to do this will result in a penalty being applied.

The work you submit must be your own individual work. Submissions will be checked for evidence of plagiarism

Feedback Format:

Written assessment feedback will be provided. Further feedback can be discussed individually as requested by students.

Additional Information for Completion of Assessment:

Creation of the .sql file

Please read the below for clear information regarding the creation of your .sql file for the assessment. This is to clear up any questions you may have in this respect.

The below must always be followed in the first instance:

“Your database solution must not be implemented or deployed to phpMyAdmin in any way other than via your SQL script file, in other words no part of your database solution can be automatically generated by third party tools. “

This means you should not use anything other than your own text file (.txt extension or other text-based file) to generate the required .sql file that contains your DDL and DML code. This includes ensuring you do not use the PhpMyAdmin import/export functions as they will truncate data and cause errors when your script is marked – you have been warned!

You must manually create the .sql file by following the below steps:

- i) Create a text file (.txt) file using Notepad++ (or another basic text editor i.e. Notepad) that will contain all your DDL and DML code (and stored procedure code if attempted);
- ii) Manually write ‘your own’ DDL and DML code and run/test it as you have learned in the workshops by entering/pasting it into the SQL tab in phpMyAdmin (see figure 1 below);
- iii) When all your DDL and DML code has been tested and working to your satisfaction through executing in the SQL tab in phpMyAdmin, ensure it is **all** pasted into your text file. At this point you will have a text file that contains all your DDL and DML code in the correct running order (DDL first of course!)
- iv) As a final check with all your code contained in your text file, copy all the code from your text file into the SQL tab again in phpMyAdmin and run it - ensure it fully creates and deploys your database, and runs all your DML statements to your satisfaction and meets the criteria you are aiming for – your .sql script must also create the database from scratch so when you run the script each time you may have to delete the database it created previously;
- v) When all the above is done and you are happy with it, rename the extension of your text file from .txt to .sql. **DO NOT USE ANY OTHER METHODS TO CREATE YOUR .SQL FILE**, if you do you are taking the risk of failing the assessment.
- vi) Your .sql file should now be ready for submission!
- vii) The delivery team will run your submitted .sql file for marking purposes through the SQL tab in phpMyAdmin.

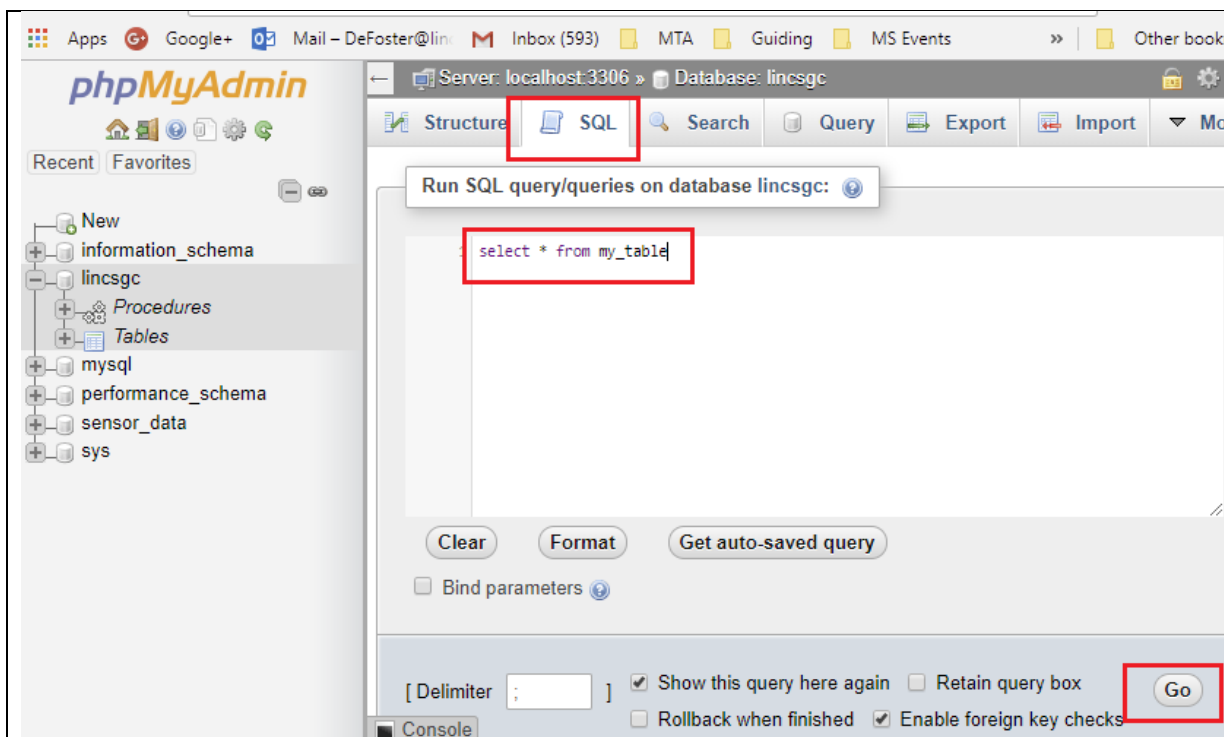


Figure 1 – SQL tab in phpMyAdmin for entering and running SQL DDL / DML statements

Assessment Support Information:

Please make sure you have a clear understanding of the grading principles for this component as detailed in the accompanying Criterion Reference Grid (CRG).

If you are unsure about any aspect of this assessment component, please seek the advice of a member of the delivery team.

Support is available in class or via Teams.

Important Information on Dishonesty & Plagiarism:

University of Lincoln Regulations define plagiarism as 'the passing off of another person's thoughts, ideas, writings or images as one's own...Examples of plagiarism include the unacknowledged use of another person's material whether in original or summary form. Plagiarism also includes the copying of another student's work'.

Plagiarism is a serious offence and is treated by the University as a form of academic dishonesty. Students are directed to the University Regulations for details of the procedures and penalties involved.

For further information, see www.plagiarism.org



UNIVERSITY OF
LINCOLN