Dalhousie University
Faculty of Computer Science

CSCI 2141 **Intro to Database Systems** **Winter-2018**

GROUP PROJECT **[80 marks]**
Submission Deadline: THURSDAY, 27th MARCH 2018

INTRODUCTION:

This group project allows you to showcase your database knowledge, as well as enhance your team working and independent learning skills. You will implement a database system and its front-end application by choosing one from the list provided in this document or proposing your own. You will be designing and creating:

1. A database, using MySQL, to store data relevant to the application domain
2. A front-end application, using any programming language or front-end tool you are comfortable with (e.g. Java, Visual C#/C++, PHP, Python, Power Builder, Visual Basic, …). The application should connect to the database and be able to manipulate the data in the database. Users of your application should also be able to query the database for information that should be presented in the form of lists or reports.

Your tasks for this project include:

- Listing requirements and business rules (you may explore a similar existing system to come up with a list of requirements for your database and its front-end application)
- Designing the database by identifying the entities and attributes from the list of requirements and business rules
- Writing the scripts to create the database in MySQL and populating it with meaningful sample data
- Creating a front-end application that makes use of this database by allowing a user to add, modify or delete data, as well as to provide means to view existing data in a meaningful way (through lists or reports).

Note: Please remember that the frontend application you create is meant for ordinary users who generally have no idea what SQL is, and how to write queries. (For example, think of the grocery store creating invoice for your purchases).

What you need to submit:

(Note: Only one submission per group)

You are required to upload the following in separate zip files:

- **Zip file 1:** A document (preferably MS Word, but pdf accepted) containing the following:
  - Section 1: A summary of your project.
  - Section 2: The **set of requirements and business rules** that are used as a basis for designing your database
  - Section 3: **Conceptual data model** of your database
  - Section 4: **Physical data model** of your database, which must be reverse engineered using MySQL workbench
  - Section 5: **Scripts** for your database. These should include:
    - Section 5.1: **SQL script** for creating the database, tables, references and indexes as required

- Section 5.2: **SQL script** for populating your database with meaningful sample data
- Section 5.3: **SQL script** for creating views and/or stored procedures (optional – has Bonus marks)

- **Zip file 2:** Your front-end application containing the following:
  - **Source files** for your application
  - **Binaries** (if applicable)
  - **contribution.txt** file specifying who in your group contributed to what components of the project. Please be specific. Everyone cannot contribute to everything, and one person in your group should not do everything. How you manage/distribute the workload within your group is entirely up to the group. ***Marks for your assignment may vary up to 20% within the group based on each member's contribution stated in this file. Also note that a group member who does not contribute anything to the project may get a zero mark for the project.*** Ensure that each member of your group has seen this file and agrees to their contribution. Any disputes among group members should also be mentioned in this file, and not emailed to me separately.
  - **readme.txt** file specifying which software/tools (including their versions) you used for creating the front-end application. This file should also contain any instructions necessary for compiling/running your code. *Remember that we cannot grade it if we cannot run it or see it in action. Don't use any third-party libraries that are not freely available.*

Important dates and submission deadline:

- Project start date: Tuesday, 06-Feb-2018
- Submission deadline: Tuesday, 27-Mar-2018

Grading criteria: *[80 marks + 10 bonus, not exceeding a maximum of 80]*

- ***Database and application requirements*** ***[15 marks]***
  - List of requirements and business rules (10)
  - Conceptual data model based on stated business rules (5)
- ***Database implementation*** ***[30 marks]***
  - Physical data model (10)
  - DDL scripts for creating database and tables (10)
  - DML scripts for sample data insertion (10)
  - Scripts for any views or stored procedures created (Bonus 5 marks)
- ***Front-end application*** ***[15 marks]***
  - Connectivity to database (5)
  - Data manipulation (insertion, updation, deletion) through the application (5)
  - Data retrieval through the application (5)
  - Using views and/or stored procedures through the front-end (Bonus 5 marks)
- ***Quality of work*** ***[20 marks]***
  - Quality of report submitted (5)
  - Quality of database design (Complexity and level of detail to which the database is designed and implemented) (10)
  - Quality of application's implementation (5)
- ***Teamwork and Contribution***

- Up to 5 marks will be deducted from marks obtained by each member of the group if the ***contribution.txt*** file is missing from the submission. A student who makes no contribution to the project may get a zero mark. (-5)

Grading Scale for Group Projects:

The grading scale definitions and the corresponding percentage points are in line with the Dalhousie's grading system provided at this link: https://www.dal.ca/campus_life/academic-support/grades-and-student-records/grade-scale-and-definitions.html

	Excellent	**Good**	**Satisfactory**	**Marginal**	**Inadequate**
Database and application requirements	Requirements and business rules are clearly stated and are completely reflected in the conceptual data model.	Requirements and business rules are clearly stated and most of them are reflected in the conceptual data model	Most of the requirements or business rules are stated and some of them are reflected in the conceptual data model	Effort has been made to list down requirements or business rules and a few are reflected in the conceptual model	Requirements or business rules are minimal or missing. Conceptual data model does not adequately reflect the stated rules and requirements
Database implementation	Physical data model expands on the conceptual data model and completely captures the requirements. Shows all attributes with their data types, field lengths and conditions (if any). All scripts are provided for creating the database and tables, and for populating them with meaningful data. Scripts execute without any errors in MySQL	Physical data model expands on the conceptual data model and captures most of the requirements. All scripts are provided for creating the database and tables, and for populating them with meaningful data in important attributes. Some random data in attributes not commonly used is present. Only minor syntax errors in scripts. Most of the scripts execute without any errors in MySQL.	Physical data model is provided and expands on the conceptual data model. Scripts are provided for creating databases and tables, and for populating them with random data. Only minor syntax errors in scripts. Most of the scripts execute without any errors in MySQL.	Physical data model is provided. Some scripts to generate database and tables are missing, or scripts for populating tables is missing. Syntax errors in several scripts prevent their execution in MySQL.	Physical model is missing or provided model doesn't reflect the requirements. Scripts for database and table creation are missing or execute with errors in MySQL.
Front-end application	Application can connect to the database without errors, and provides means to add, modify and delete data from relevant tables. Application also provide means	Application can connect to the database without errors, and provides means to add, modify and delete data from most of the relevant tables. Application also	Application can connect to the database without errors, and provides means to add, modify and delete data from some of	Application can connect to the database without errors, however limited functionality is provided by the front-end	Application cannot connect to the database; or very limited or no means to add, modify or delete data, or to view existing

	to view existing data from relevant tables (as lists or reports for example).	provide means to view existing data from most of the relevant tables.	the relevant tables. Application also provide means to view existing data from some of the relevant tables.	application to add, modify and delete data or to view existing data from relevant tables.	data from the database through the front-end application
Quality of work	Submission exactly as per submission instructions. Database design is sufficiently complex that it captures more than just the basic functionality present in similar projects. Application interface is well thought out with data clearly presented to the user. Report submitted is clear and to-the-point.	Submission as per submission instructions. Database design captures the basic functionality present in similar projects. Application interface is generally well thought out with data clearly presented to the user. Report submitted is clear and to-the-point.	Submission mostly as per submission instructions. Simple database design that captures the minimum functionality present in similar projects. Application interface is clear. Report submitted is clear and to-the-point.	Submission deviates from submission instructions. Database design does not capture the minimum functionality present in similar projects. Application interface is unclear. Report submitted is incomplete.	Submission significantly deviates from submission instructions. Poor database design. Application interface is not provided or very poorly made. Report is incomplete
Teamwork	Contribution.txt file provided that clearly highlights each team member's contribution. Marks for your assignment may vary up to 20% within the group based on each member's contribution stated in this file. A non-contributing member may get a zero mark for the project. **Deduction of 5 marks if this file is missing.**				

DATABASE PROJECT IDEAS:

Note: Implementing any of the databases provided in your textbook is not allowed.

Note: The description for databases given below are incomplete and meant to spark your thinking process only. Please do some research on similar existing applications to see what information might need to be stored in your database. The front-end application should allow you to add/modify/delete data for the attributes. It should also allow you to provide listings and reports for the data stored in the database.

- **Project 1:**
  Build a database for an online multiplayer game (e.g. World of Warcraft, Tanki online for PC games; War Dragons, Avatar Fight for mobile games, …). The database could store information about the players, teams, game characters, game items, …

- **Project 2:**
  Build a database for an Entertainment TV channel (e.g. CBC, BBC, ABC, …) that telecasts movies, documentaries, series, gameshows, talk shows etc. The database could store information about the programs, their schedules, their cast, the show hosts, …

- **Project 3:**
  Build a database for an online room rental agency (e.g. point2homes.com, property.trovit.ca, viewit.ca, …) that provides room rental for individual or groups of

university students. The database could store information about student residents, their requirements, rooms available, facilities available both inside the rooms as well as shared, distance from the student's department or campus, their means of available transport, …

- **Project 4:**
  Build a database for an online vehicle rental agency (e.g. avis.ca, budget.com, enterprise.ca, …) that provides vehicle rental for individuals or companies. The database could store information about customers, vehicles and their details, features desired or available in vehicles, insurance information, vehicle availability, rentals, …
- **Project 5:**
  Build a database for Halifax transit. The database could store information about the vehicles, drivers, mechanics, support staff, schedule, services provided, bus routes, …
- **Project 6:**
  Build a database for Dalplex. The database could store information about registered users, equipment, events, schedules, parking, employees, …
- **Other projects:**
  Think about how various services like Police, Emergency Health Services (or Paramedics), Roads, Waste Management, Forestry, … work. What database could you provide that will implement the services that they currently offer to the public or help in improving their services. How these services keep track of their assets and facilities, …

Choose one from above, or think of interesting uses and applications of databases. Design and implement how you would like them to be.

Useful links for ODBC/JDBC connectivity to MySQL:

Programmers developing frontend applications can connect to databases using a variety of methods. For example, applications developed in JAVA can connect through JDBC, those developed in Visual C++/C# can connect through ODBC. Some useful links that explain how to connect to a MySQL database are given below:

JAVA to MySQL connectivity:

- https://dev.mysql.com/doc/connector-j/5.1/en/connector-j-usagenotes-connect-drivermanager.html
- http://stackoverflow.com/questions/2839321/connect-java-to-a-mysql-database
- https://www.tutorialspoint.com/jdbc/jdbc-sample-code.htm

Visual Studio to MySQL connectivity:

- https://dev.mysql.com/doc/visual-studio/en/visual-studio-making-a-connection.html

PHP to MySQL connectivity:

- https://www.w3schools.com/php/php_mysql_connect.asp

Python to MySQL connectivity:

- https://dev.mysql.com/doc/connector-python/en/connector-python-example-connecting.html

Several YouTube videos and tutorials are also available that show how to connect you choice of application development environment to MySQL database.