DATABASE: CASE STUDY
FINAL CAPSTONE PROJECT

For your case study complete the following tasks:

- Part 0: Overview
  - (onset) Provide a **cover letter** to the company you are "working for" from "your own company" introducing yourself. Use this letter to convey your **elevator pitch** for the project.
    - *Hint: Imagine you are speaking with Mr. Dorsel, he wants to know the big picture but may decide to leave the details to others.*
  - (conclusion) Provide an **executive summary** of your solution. This document is a more detailed description of how you solved the problem. Do not get caught up in how you phrased the SQL (that will come later).
    - *Hint: Imagine you are speaking with Mr. Shannon, he knows the technology so will understand the jargon, but does not want to know (at this level) the minutia of how you executed specific tasks.*

- Part 1: Analysis
  - Complete the **ERD**
    - identify all candidate keys and surrogate keys
    - identify the primary key
    - identify all foreign keys
    - identify the relationship between tables (one-to-one, many-to-one, many-to-many, etc) and represent them in your final ERD
  - compile a list of **functional dependencies** as A,B → C, D for each table
    - from the functional dependencies declare each relation, eg TABLE01(**A, B**, C, *D*), where A and B are composite keys, and D is a foreign key.
  - demonstrate each that relation is in **BCNF** (see http://www.ict.griffith.edu.au/~jw/normalization/index.html)
  - as a self-check **generate** the ERD within mySQL Workbench and confirm that both diagrams agree

- Part 2: Create and populate each table
  - identify all **integrity constraints** and confirm that they are being upheld
  - write the **SQL** to **create** the tables
  - write the SQL to **populate** the tables
    - use the Dummy Data for mySQL Database website (http://filldb.info/)
      - be sure that key **constraints** are maintained!
    - be sure to include a few entries that properly reflect the case you are working on. For example, if there is a requirement that a president of the firm be identified then do so.
  - Submit the SQL to create the tables and the 1st 10 SQL statements that load each table.

- Part 3: Queries
  - write and test the **basic 10 queries** that were identified for your case study (eg https://www.db-fiddle.com/ is a great place to test your ideas)
  - demonstrate the SQL answers the question posed

- Part 4: Implement your solution. AFTER YOU HAVE CONFIRMED EVERYTHING IS CORRECT!
  - **Load** your solution onto a mySQL server. You have been assigned a schema on the classroom server. Login with the primary schema named CASE0#_*ID* where # is your particular number and ID is the unique character sequence of your case.
  - test that your solution functions as anticipated
  - **backup** your schema (i.e. ready your solution to be ported to the teacher machine for grading)

- Part 5: Programming
  - write an application in any **language** you wish to present a user with a front-end to your queries. A simple menu interface will suffice
  - generate well formatted reports with clear titles, labels, etc

- include in your solution the ability to display any table (i.e. in mysql **select * from *sometable* limit 10;**)
- submit your well **documented** code and HOWTO.txt and/or README.txt files with your final packet
- *IT IS NOT SUFFICIENT THAT USERS ACCESS THE FULL MYSQL TOOL TO RUN YOUR QUERIES! THEY SHOULD NOT BE GRANTED THAT LEVEL OF ACCESS!*

- Part 6: Present your solution *to the Board*
  - imagine you have completed the task and must now go to the company and describe what you did. Prepare a **presentation** that will show people what you did and how to access your solution.
  - submit your presentation along with other materials.

- Part 7: Reflection
  - PMI (plus-minus-interesting) was the case interesting? Did it help to solidify the course concepts?
  - How long did it take your team to accomplish the task?

- Part 8: References
  - an APA list of websites, texts, etc that were used in the process of solving the case.

- BONUS
  - write your solution to run on the **web** using LAMP or equivalent.
  - limit access to your database by invoking **views**
  - you suggest and implement an **enhancement** to this project (*must be approved by the instructor*).
  - as part of the application, implement the ability to process a **transaction** (e.g. rent a car, assign a room)

- CHANGES
  - the case and support materials should be sufficient to complete the task. However, should you need to change the tables etc, you must submit a **CHANGE ORDER** and get approval before you move forward. A CHANGE ORDER is a description and rationale why you feel the change is warranted. All such documentation should be included in an Appendix to your final document.

PAGES/DELIVERABLES:

- ☑ Cover letter
- ☑ Executive summary
- ☑ Completed ERD
- ☑ List of keys
- ☑ Functional dependencies and resultant BCNF tables
- ☑ Proof of BCNF
- ☑ SQL to create tables and enforce integrity constraints
  - Well commented
  - 1st 10 SQL statements that insert data into table(s) is sufficient
- ☑ SQL for 10 queries supplied with the case
  - Well commented
  - Include up to the 1st 10 lines of output, with clear column headings
- ☑ Schema up and running on mySQL server
  - Export the database to a dump file using mysqldump or similar backup feature
  - Generated ERD within mySQL Workbench
- ☑ Application
  - Well documented program
  - Includes menu for the 10 case queries
  - Includes menu option to allow a dump of any single table
  - Screen shots
  - Well formatted reports
  - README.txt/HOWTO.txt
- ☑ Presentation
- ☑ Reflection paper
- ☑ Reference page(s)
- ☑ Appendices to include addendum (as needed)
  - Bonus material attached here
  - Change orders attached here
  - Corrections, modifications to the case study

The case study description and this document provide sufficient material to get started. However, the instructor reserves the right to update and modify the case requirements as needed.