Designing a database for course lists and competencies

Group Project

Students will work in groups of 3 to design and develop a database and its associated Java application[[1]](#footnote-1). The source code must be tracked in a private GitLab repo shared with the instructor. The project must meet the requirements outlined below.

# Learning Objectives

* Analyze and design a relational database from the given problem.
* Use normalization and ERDs to communicate database design
* Create PL/SQL packages and programs to support an application.
* Create a user facing Java application that allows the user to interact with the database.
* Work in teams to implement a database and application.

Contents

[Learning Objectives 1](#_Toc104536078)

[Project Overview 1](#_Toc104536079)

[Project Requirements 1](#_Toc104536080)

[Problem 1](#_Toc104536081)

[Courses 1](#_Toc104536082)

[Competencies 2](#_Toc104536083)

[Database Requirements 2](#_Toc104536084)

[Technology 2](#_Toc104536085)

[Data 3](#_Toc104536086)

[Application Requirements 3](#_Toc104536087)

[Technology 3](#_Toc104536088)

[Deliverable (due November 18th) 3](#_Toc104536089)

[Resources 4](#_Toc104536090)

# Project Overview

The project will consist of a database designed and implemented on PDBORA19C and a Java application that interacts with the database. The database will track computer science courses, their pre and co-requisites, and the competencies associated with the course. Students will be provided with a source dataset to be transformed into a proper database design. A Java application will then be designed to facilitate creating, reading, updating, and deleting information from the database.

## Problem

To provide better data integrity, the computer science department wants to use a database to keep track of its courses, how they relate to each other, and how ministry competencies are covered by these courses. Today, this information is documented in spreadsheets, on our website, and in pdfs. Your objective is to examine the data, design an appropriate database, using normalization, and create a Java application to facilitate interacting with the data. You will be provided with the data sources and with a brief explanation of what this information means.

### Courses

A program such as computer science, has an associated code and is made up of courses. The courses can either be program specific or be part of general education. Each course has a unique identifier, a name, description, and ponderation. Analyze the courses for 420.B0 Computer science.

Some information associated with a course is derived from other information. For example, the total number of hours is the sum of the classroom and lab hours for 15 weeks. However, the total number of credits is also calculated. Classroom, lab, and homework hours over 15 weeks determines the total number of credits per course, where 1 credit is equal to 45 hours. Note, the number of credits can be fractional. For example, a course of 3-2-3 ponderation would be worth 2 2/3 credits.

The list of courses can be found here <https://www.dawsoncollege.qc.ca/computer-science-technology/course-list/> or in the provided PDF.

Courses can also be a part of a particular domain. A domain also has a description. The list of domains and their associated courses can be found here <https://www.dawsoncollege.qc.ca/computer-science-technology/domains/> .

### Competencies

To ensure courses have consistent content and are of high quality, courses are designed to meet certain competencies as described by the Ministry of Education.

Each competency has a statement and some goals to be achieved. Competencies are made up of elements. Each element has a title, some criteria, and an order of where it appears in the competency. For more information on how competencies work, look at the Program Specific Component of the PE01ComputerScienceTech-420B0-2018.pdf. This documents all the competencies of the 420.B0 program. Additionally, a glossary of these terms can be found in the Additional Information section of PE01ComputerScienceTech-420B0-2018.pdf.

There are many rules around competencies and how they are implemented in each program. The computer science department uses the FinalGrid.xslx spreadsheet to keep track of what course teaches which competency. More importantly, the program must keep track of which element of the competency is being taught by a given course. Additionally, the number of hours associated with each competency must be tracked. Finally, some courses indicate if a competency is terminal, where a competency can terminate only once for the entire program.

For example, a course does not need to cover all elements of a given competency. However, if a competency is included, all elements of the competency must be covered across the program. Additionally, a course cannot mix general competencies with specific competencies. Finally, a course might terminate a given competency.

# Project Requirements

## Database Requirements

### Technology

* The database must be developed using PL/SQL and be runnable on PDBORA19C.
* The database must be designed in a fully normalized way, avoiding issues related to data anomalies
* The subprograms should be logically grouped into one or more packages.
* There must be an installation subprogram that initializes the database.
* There must be an uninstall subprogram that removes the database.
* PL/SQL subprograms should be documented (commented) in a way equivalent to Javadoc

### Features

* Procedures to interact with the database
  + Procedures must be designed to allow the application to interact with the database. The application must NOT be able to modify, select, or manipulate tables directly
* Functions to compute derived data and perform validation
  + Functions must be designed to return information to the applications interacting with the database. These functions should compute the derived values can contain the business logic.
* Record user activity for auditing purposes
  + To ensure data integrity, modifications to tables must be tracked

### Data

The database must come with some information already entered. Sufficient data should be present in the database to ensure that all features can be tested. However, it is not required to enter every course, competency, and element in the database.

## Application Requirements

### Technology

* The client application must be developed using the version of Java used in your programming courses.
* The client application must connect and interact with the Database using JDBC (the java.sql package) provided in Java.
* The application must be implemented using separation of concerns.
* A private GIT repository must be used for source code management.
  + The instructor must be added as a maintainer to the repo.
* Java classes and methods should be appropriately documented using the standard Javadoc style.

### Features

The application should have an interface for the user (command line or GUI) that provides the following features:

* Login to the database
  + Users must be prompted to provide their username and password
* Retrieve and display data from the database
  + Users should be able to display information related to courses and competencies
  + This information should be presented in a user-friendly way, i.e., not just a dump of the tables as they appear in the DB
  + This should include derived information, for example the total number of hours for a course, or its number of credits, etc.
* Add data to the database
  + Users should be able to enter information related to a new course or competency
  + The information should be provided in a user-friendly way, where the user does not need to understand the complexities of how the tables are designed in the DB
* Update data
  + Users should be able to update an existing course or competency
* Delete data from the database
  + Users should be able to delete an existing course or competency
* Validate data
  + Users should be able to call a function to validate if courses and competencies follow the rules described above. A list of issues should be reported to the user.
* Display audit log
  + Users should be able to display the logs tracking modifications to the tables

# Deliverable (due November 18th)

Submit a zip called GroupProject.zip through Moodle of your Git repo. The following files must be included in your zip (and your repo).

* Readme.md
  + Include a link to your Gitlab repo
  + Ensure all steps to setup the database is well documented
  + Ensure all steps to compile and run the Java application are well documented
  + Ensure names and student ids of group members are listed
* Design.pdf containing ERD diagrams describing your final design
* Source code for the PL/SQL packages
  + setup.sql which setups up the database, creates the appropriate packages etc.
  + remove.sql which removes any tables, types, and packages created by setup.sql
* Source code for the Java application
  + Must include any libs or jars required to run the application

In addition to the final deliverable described above, lab activities will be provided to assist with the project. All group members will be expected to be able to discuss and demonstrate their progress in the project during class time. For due dates of specific deliverables please see Moodle.

# Resources

* Dawson college Computer Science 420.B0 course list: <https://www.dawsoncollege.qc.ca/computer-science-technology/course-list/>
* Dawson college Computer Science Domains list: <https://www.dawsoncollege.qc.ca/computer-science-technology/domains/>
* Competency mapping: FinalGrid.xslx
* Competency descriptions: PE01ComputerScienceTech-420B0-2018.pdf

1. Assignment 7 and Assignment 8 have laid the groundwork for the project and students should reuse code, designs, and concepts seen in those activities. [↑](#footnote-ref-1)