# CPSC437 S24 - LAB 1 Report

Freeman Irabaruta

February 13, 2024

## 1    Introduction

This report details the design choices and functionality of the code developed for CPSC437 S24 - LAB 1. The assignment involved two main tasks: aggregating department salary statistics and identifying overlapping sections in a university database.

## 2    Task 1: Department Salary Statistics

The goal of Task 1 was to aggregate salary statistics by department and visualize the results.

The aggregation approach involved querying the instructor table for department names and salaries, and then calculating the median, average, and standard deviation of salaries for each department (lines 53-69 in 'task1.py').

The results were then plotted using matplotlib and saved as a PNG file (lines 77-93 in task1.py), and written to a CSV file and the salary statistics table in the database (lines 96-112 in task1.py).

## 3    Task 2: Overlapping Sections

Task 2 involved identifying overlapping sections in the university database.

The table design choice for the overlapping sections table included fields for the day, course ID, section ID, year, semester, and overlap start and end times for both overlapping sections (lines 37-53 in task2.py).

The algorithm for identifying overlaps involved comparing the start and end times of each pair of sections that occur on the same day, semester, and year (lines 100-117 in 'task2.py'). The results were then written to the overlapping sections table and a CSV file (lines 122-139 in 'task2.py').

# 4 DatabaseConnection Class

The 'DatabaseConnection' class, provided in 'database-connection.py', serves as a context manager for database connections. It simplifies interactions with the database by automatically establishing a connection upon entering a context and responsibly closing it upon exit, even in the event of errors. This class abstracts the intricacies of database connectivity, allowing for concise and clean execution of SQL commands using a cursor object that is created and managed internally. One potential improvement could be the addition of error handling mechanisms to provide more informative error messages when database operations fail.