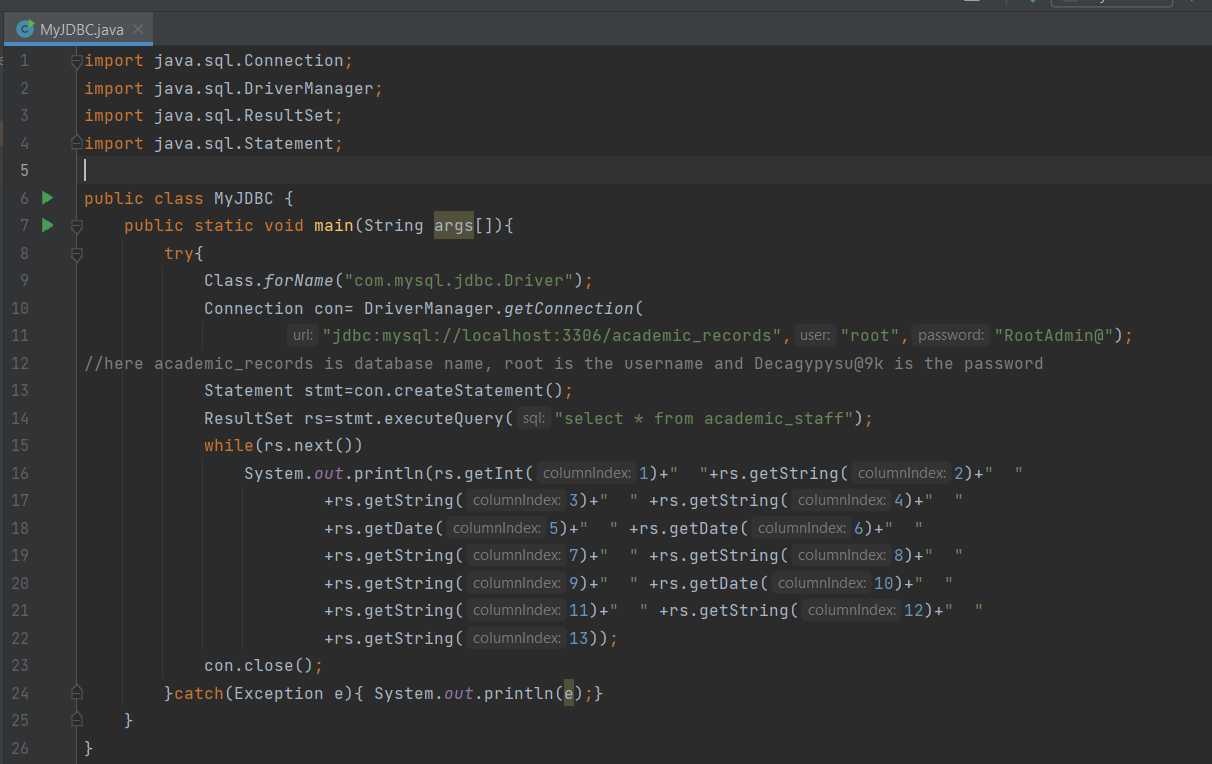
**COSC 4606 Project 2**

**Report**

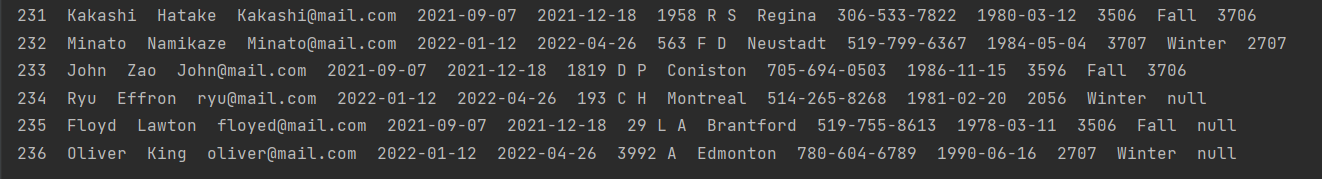
**Darrel Asare**

Screenshots for Functional Requirements

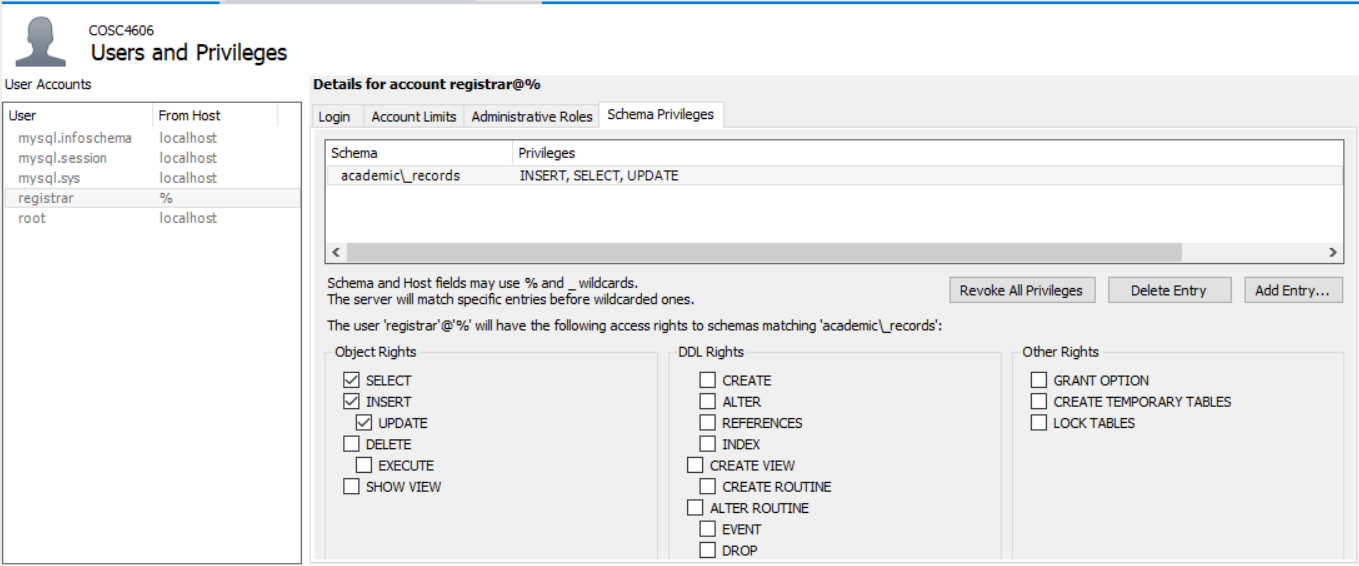
Testing java code for database administrators using root.



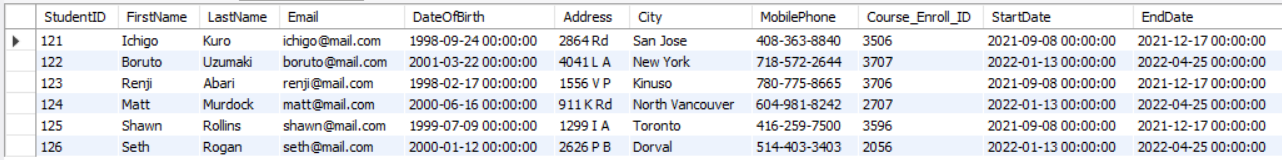
Academic\_staff



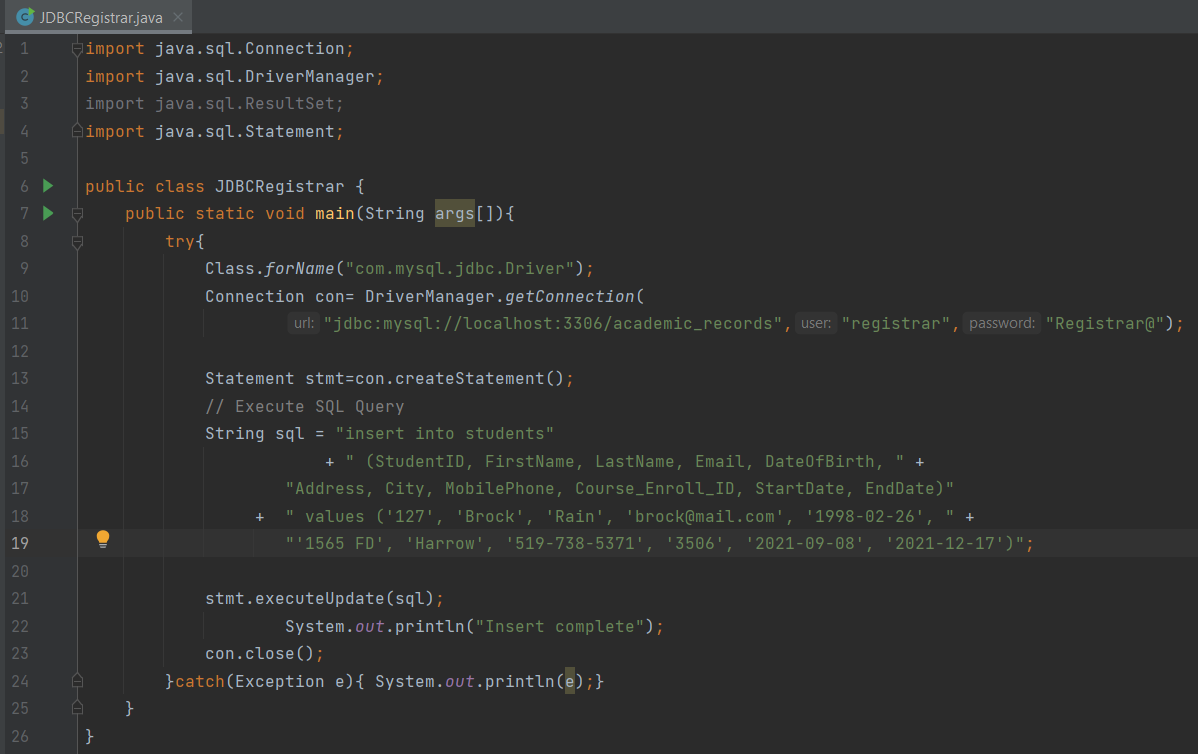
Registrar privileges

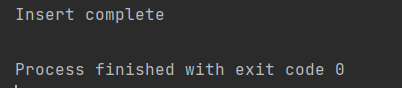


Original students table

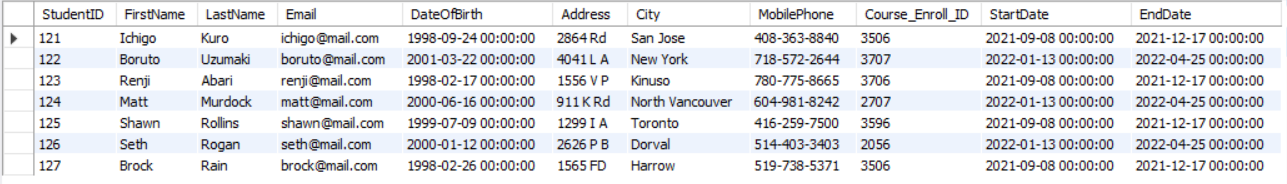


Testing java code for registrar inserting values

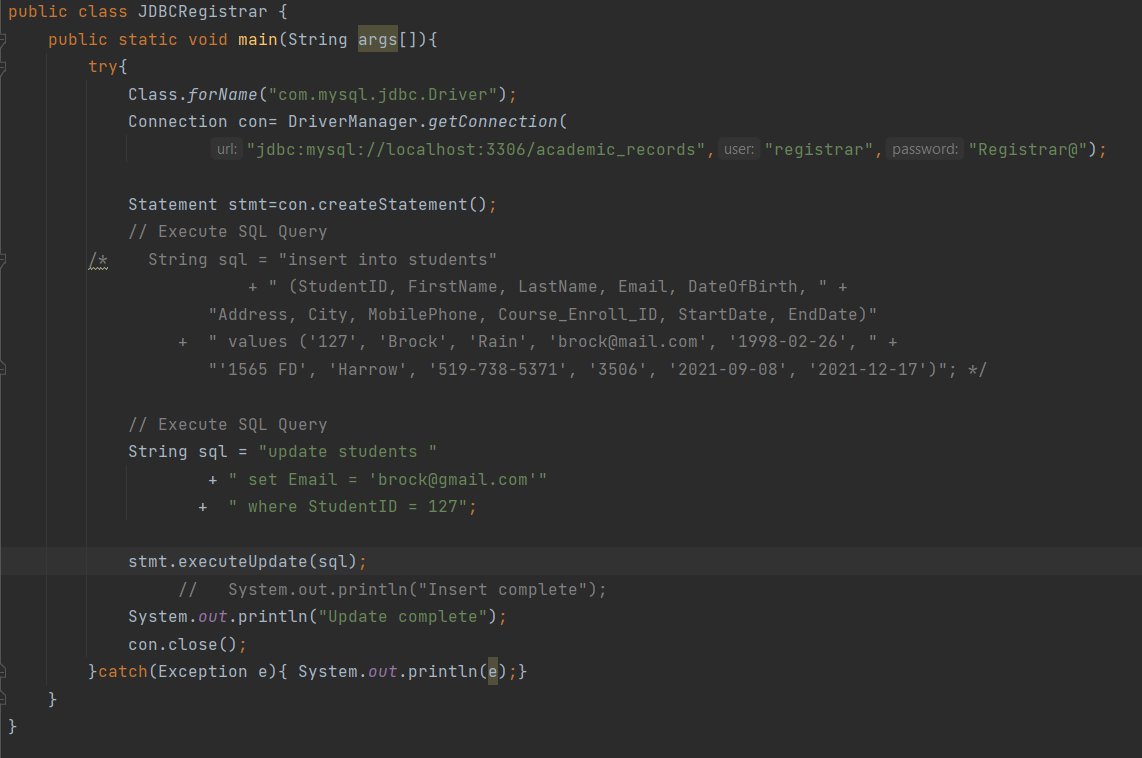


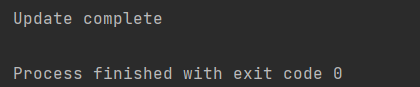


Students table after insert

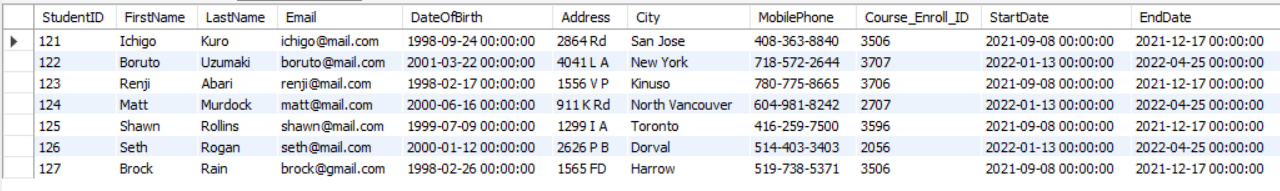


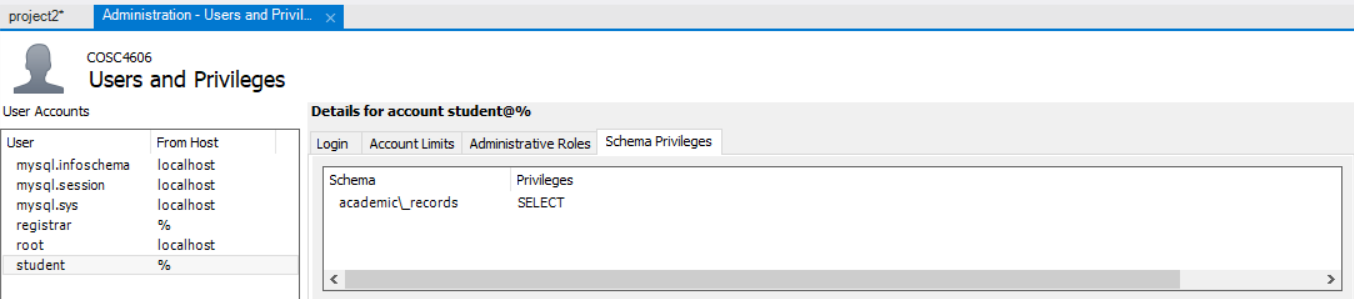
Java code for Registrar update to students table



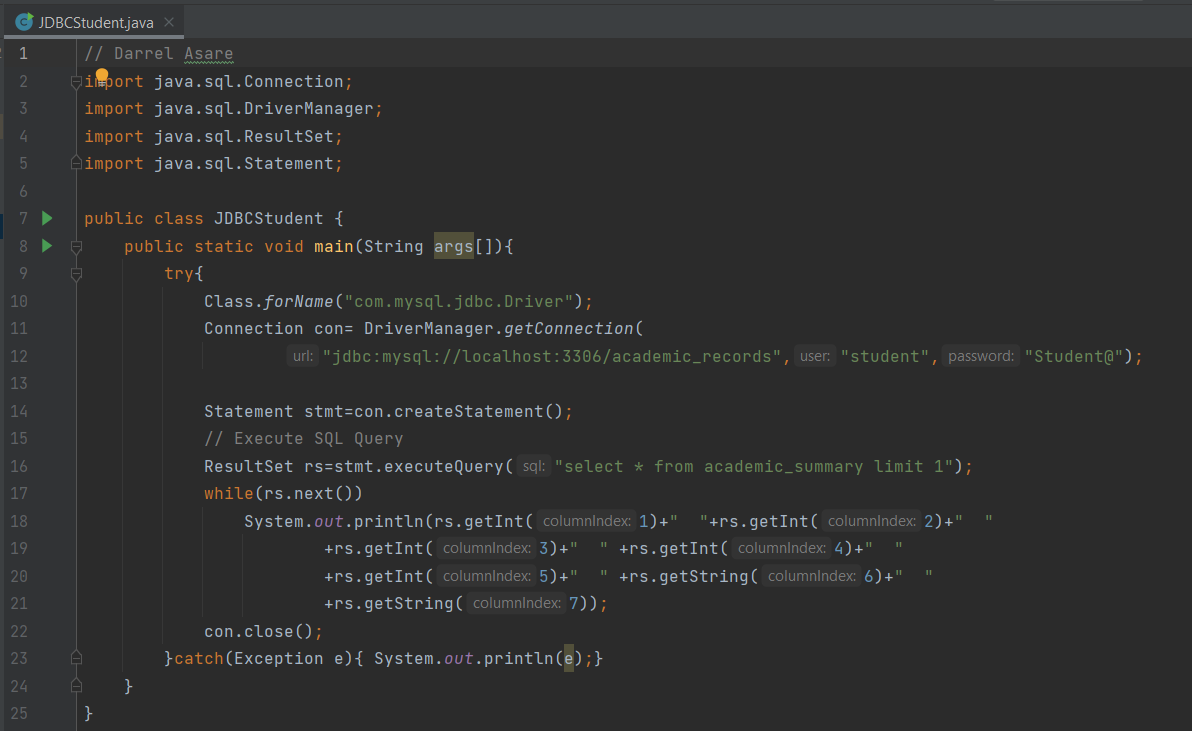


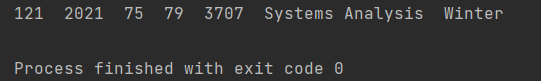
Students table after updating email to [brock@gmail.com](mailto:brock@gmail.com) instead of brock@mail.com



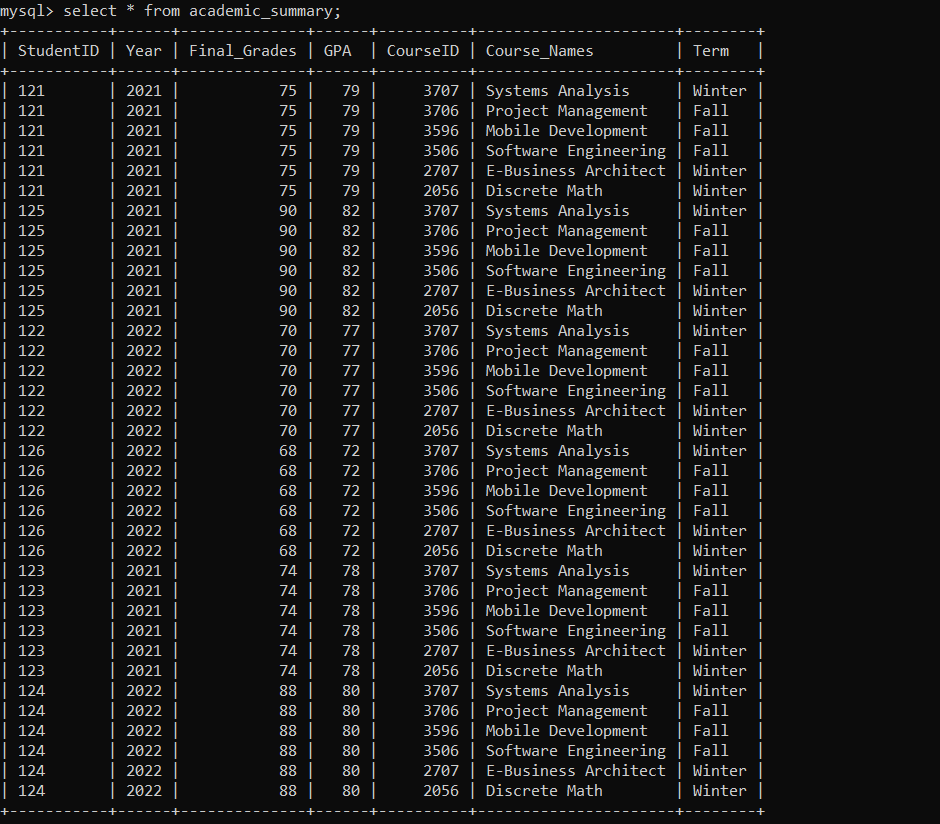


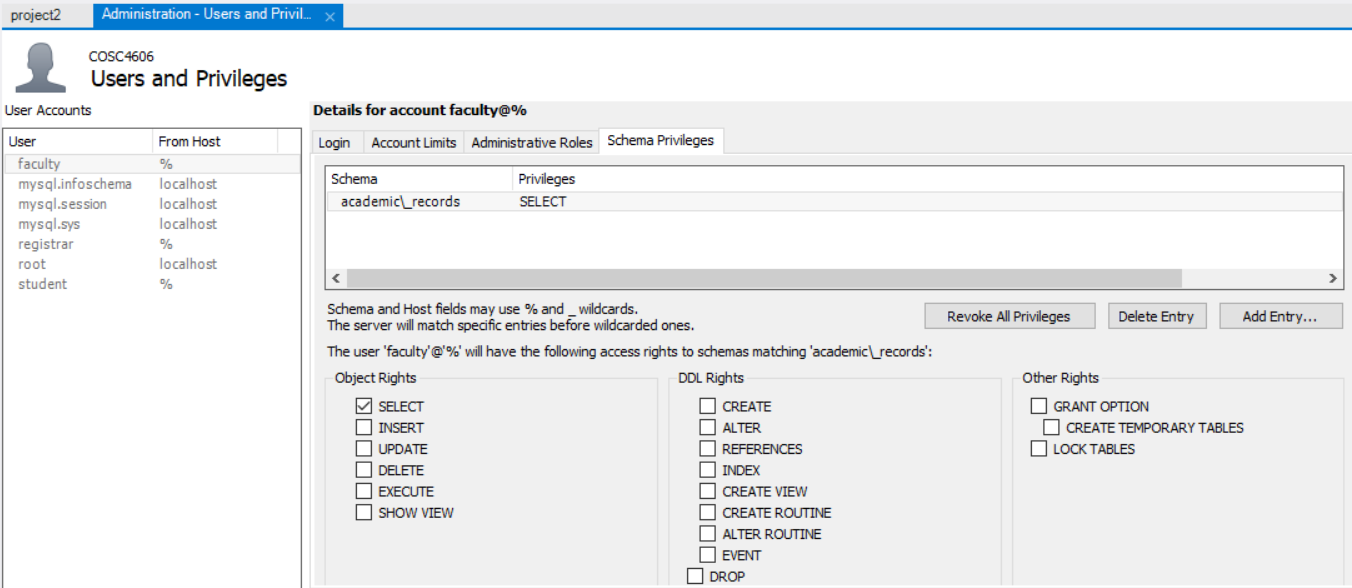
Java code for student to check academic summary



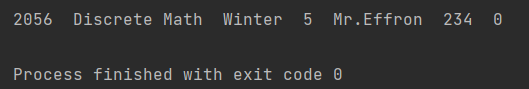
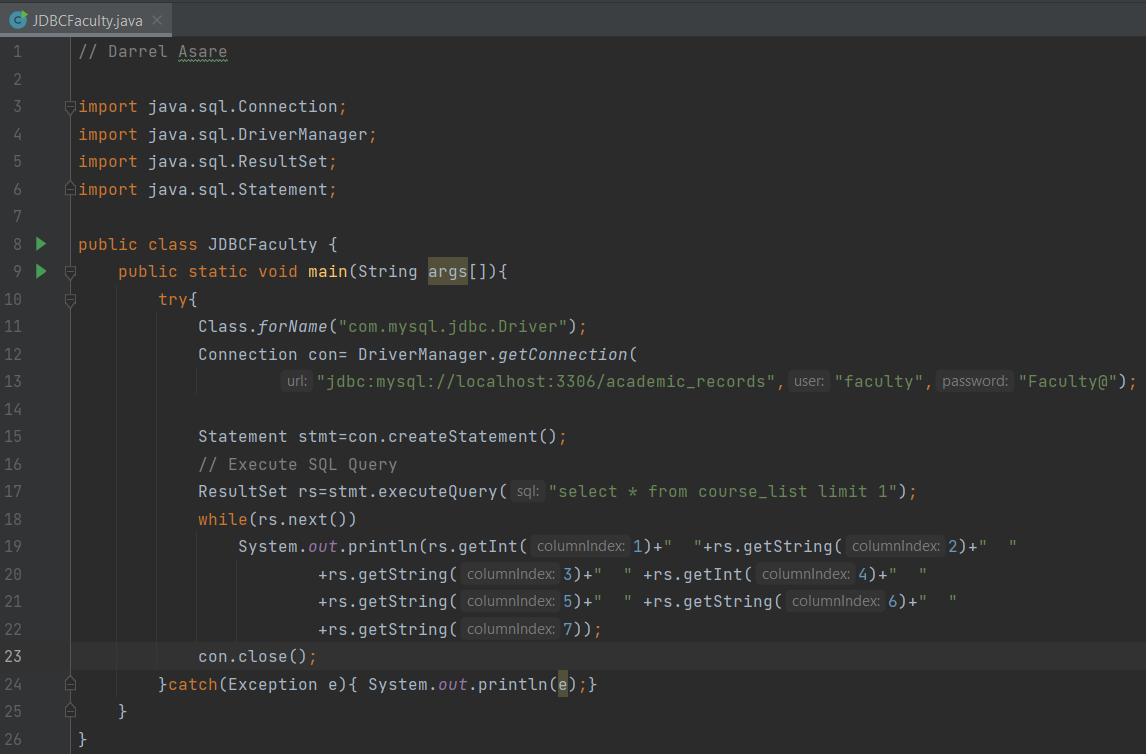


Academic\_summary table view

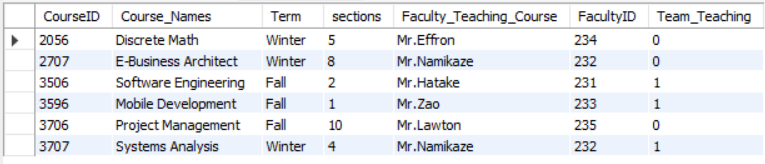




Java code for faculty to check course list he is teaching



Course\_list table view



**Forms, Reports and Deliverables**

User Manual

This report will explain the project of an academic record system using pictures and paragraphs. This time the tables and forms will be shown from SQL Workbench working with SQL Server.

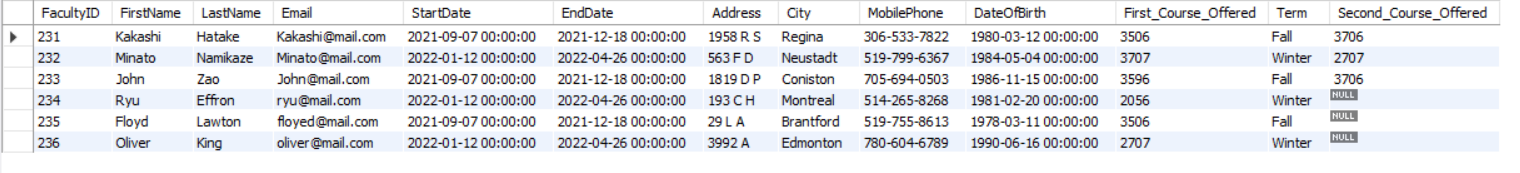
# Evidence of normalized tables

All the tables are in 1NF because each attribute in a relation has atomic single values. In the Co-op table, the student name attribute also has the student first names underscore last names. We also have a student course deletion table which is in 2NF because we moved fields from the courses table which means they do not have to be fully dependent on the primary key. This makes it easier to see which student has deleted a course instead of adding that information to the courses table, which will cause the table to increase in size.

# Printouts of test runs for forms and reports with table and query structures

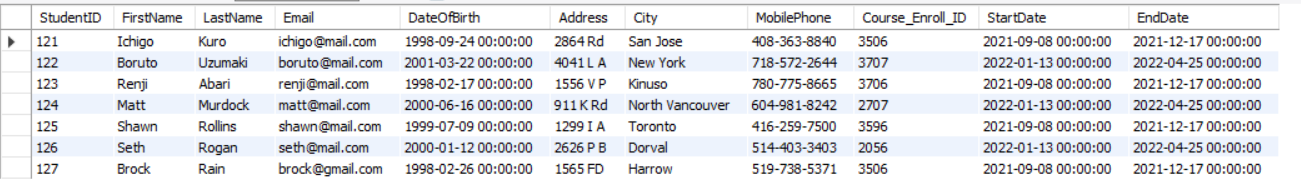
Forms

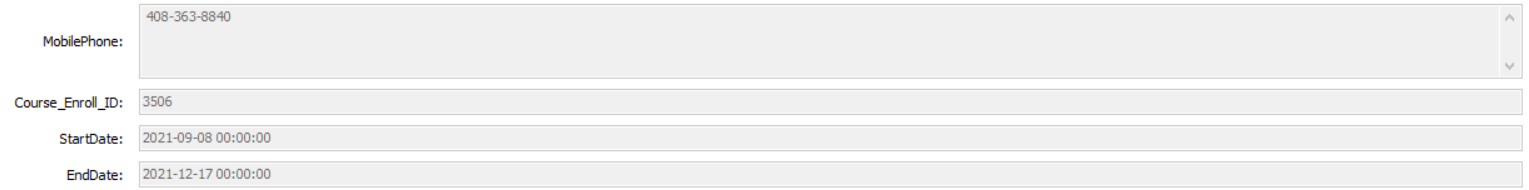
1. add a faculty member to the academic staff



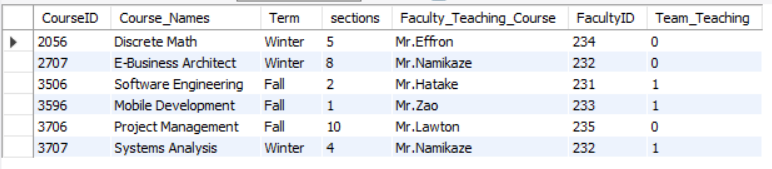
 

1. register a student in the university



1. assign a faculty member to teach a given section of a given course in a given term, permitting the possibility of “team teaching” with more than one faculty member for a given section

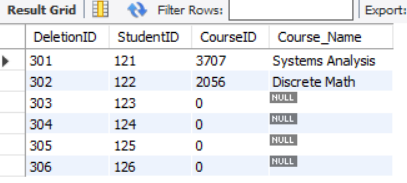




1. enroll a student in a particular course

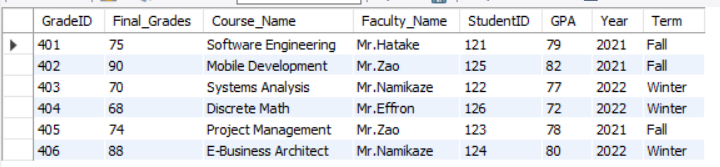
Shown in the students table with course enroll id.

1. delete a student from a particular course





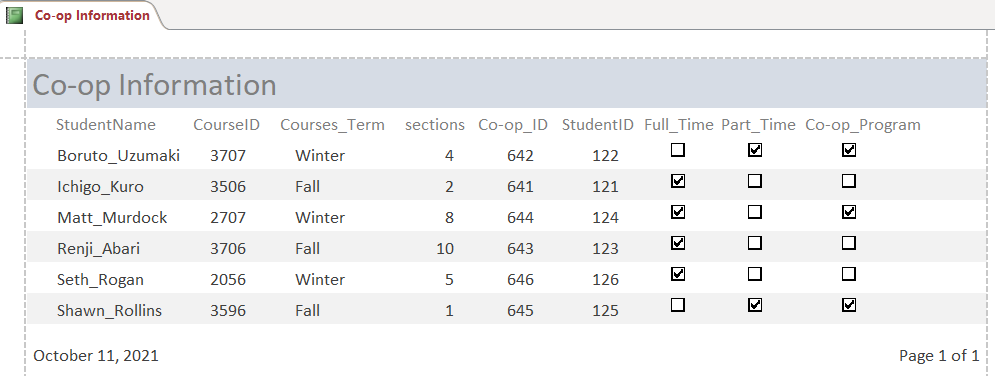
1. alter a student record to permit, for example, entering a grade for a completed course



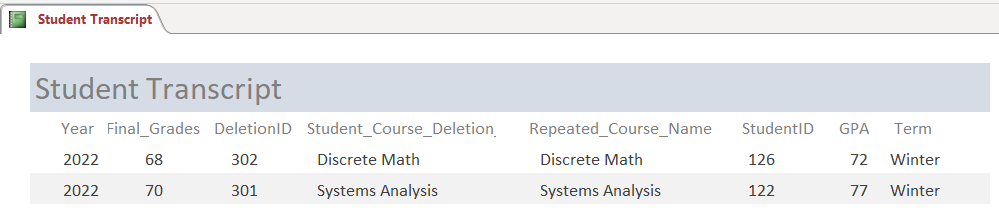


Reports and Information from project 1

1. display a class list, given a course id, term and section id, sorted by student name. Indicate whether each student is full-time or part-time that term and whether the student is in the co-op program.

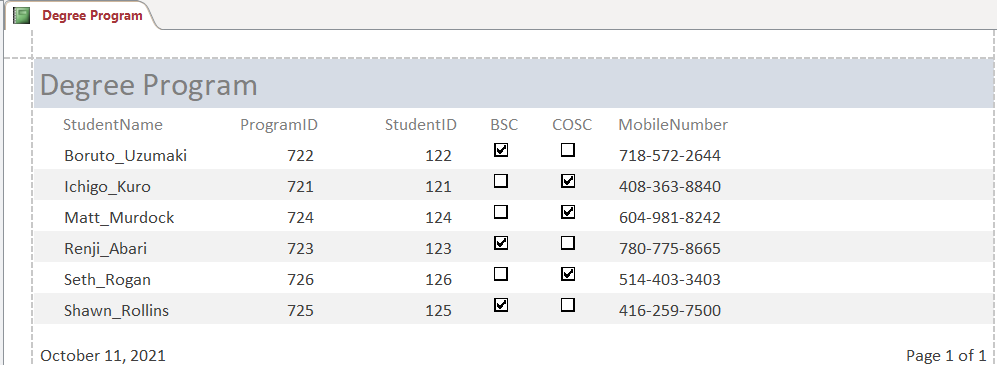


1. display a student transcript, given a student id. Show all courses enrolled, including failed courses and dropped courses, with an average grade displayed for each academic term and the average overall, using in the latter case only the best mark for repeated courses. The transcript should be ordered by academic year, earliest to latest, and within a given year by course id.

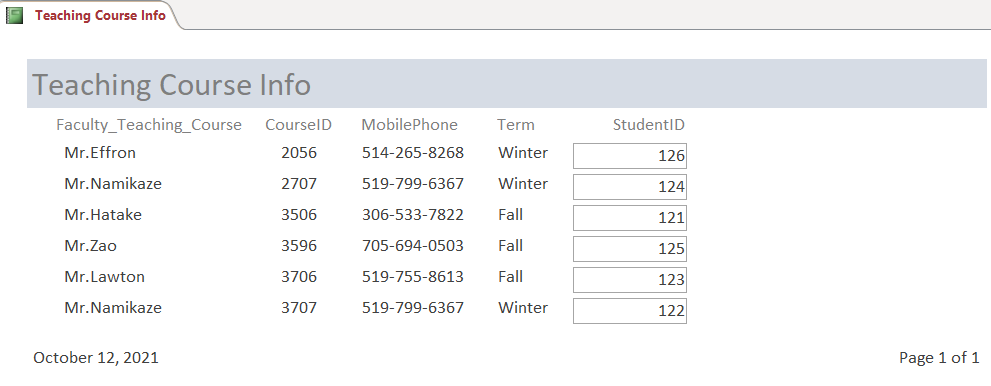


Final grades after they repeated the deleted course.

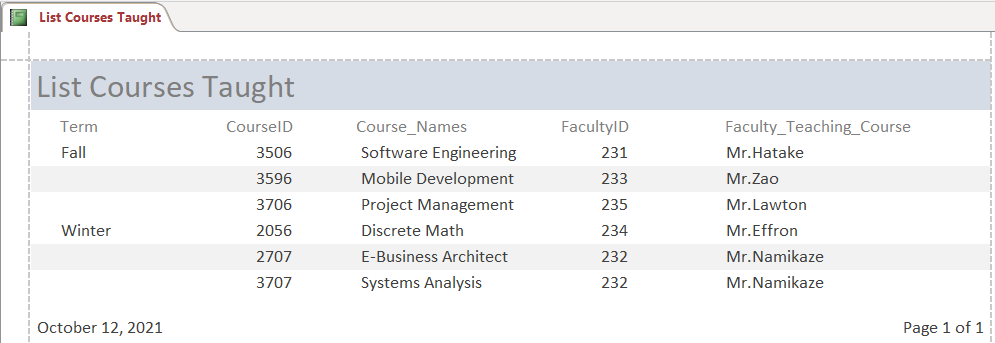
1. display a table of all students (name, id and contact phone number) currently enrolled in a given degree program (eg. B.Sc.4 COSC), sorted by student name. (Provide sufficient data to be able to run several tests)



1. display the instructor name, course id and home phone number of all instructors teaching a course to some student in a given term, given the student id and term.



1. display a list of all courses taught by a faculty member, grouped by term (earliest to latest) and sorted by course id, given an instructor id and either a term, a range of



# Database design for ER diagrams and database schemas

