

## Assignment 1 [Due on 29/01/2023]

### Instructions

1. Download and install the latest version of PostgreSQL from [here](#). For interactions with the PostgreSQL server, you can use the command line client ([psql](#)) or the GUI client ([pgadmin](#)). See [here](#) for tutorials on Postgres.

### Exercises

In this assignment, you will learn how to access and execute database queries using JDBC. You will use a university database from [here](#). Create a database named univdb. Create the tables using the given DDL file and then populate the tables using the data in the file **largeRelationsInsertFile.sql**. [You can do these directly. No need to use JDBC].

To understand the schema, you may also refer to the E-R diagram in Slide 39 (6.43) [here](#). For the following exercises, it would be more user-friendly if you can write Java code that does input/output using Web or GUI. But if you don't know, for this assignment you can do them using the Java command line interface. You can use Netbeans or Eclipse IDE.

**[20 points] Exercise 1:** Write a Java method that takes the name of a table and row count k as input from the user, and then prints the table with appropriate names as column headings and any k rows of the table. You need to create a well formatted tabular output that looks like a table. [Hint: Use JDBC metadata feature, LIMIT k clause]

**[15 points] Exercise 2:** Write a Java program using JDBC that takes course\_id from a user and then prints all the prerequisites of the given course. Print not only the direct prerequisites of the course, but also prerequisites of prerequisites, and so on. You need to print the course\_id and title of all the prerequisites.

**[15 points] Exercise 3:** Create a trigger to enforce the constraint: "An instructor cannot teach two different sections in a semester in the same slot". [Note: the constraint can be violated by changes to the teaches relation as well as the section relation]

**[20 points] Exercise 4:** Write a Java program that takes as input the roll number of a student and then prints the CGPA of the student. You can create a function to compute the CGPA given the roll number. If the input roll number does not exist, then print an error message.

Take the grades as follow:

A+	10
A	9
A-	8
B+	7
B	6
B-	5
C+	4
C	3
C-	2

**[30 points] Exercise 5:** Write a Java program that allows university administrators to print the top-k students for the following three cases: [Use the ranking() function on CGPA]

- Top-k students with highest CGPA
- Top-k students with highest CGPA in a given department
- Top-k students with highest CGPA enrolled in a given course (need to consider all the students who have taken the course across all the offerings of the course).

## How to Submit

- Create a folder as rollnumber\_firstname
- Create separate files: ex1.java, ex2.java, and so on, for each of the exercises. Put the files in the folder.
- Create an output.pdf file with the following contents:

**Ex1:** Output for department, section and takes tables.

**Ex2:** Output the course id and title of all the prerequisites for the following courses: 276, 647, 496

**Ex3:** Write any 5 varied test cases that you created to test the trigger. They should test different aspects of whether the trigger is working or not. Marks would be given based on the quality of your test cases. Please give a 1-2 line description of what each test case is testing.

**Ex4:** Print the CGPA for the following roll numbers: 76672, 90567, 4582, 81258

**Ex5:**

- Print the top-5 students in terms of CGPA
  - Print the top-5 students with highest CGPA for departments: Psychology, Elec. Eng., and Civil Eng.
  - Print the top-5 students with highest CGPA for course ids: 237, 349, 735
- Put the output.pdf in the rollnumber\_firstname folder, zip it and upload in Google classroom.