

Course Outline	2
Lectures	2
Syllabus	2
Timelines	2
Textbooks	2
Evaluation	2
TAs	3
Assignment 1 [Due on 29/01/2023]	4
Instructions	4
Exercises	4
How to Submit	5
Assignment 2 [Due on 20/02/2023]	7
Instructions	7
Overview	7
Exercises	7

Course Outline

Lectures

Timetable: E slot in A-LH2

Syllabus

[CSE-IITH UG Core Courses](#)

1. Advanced SQL (Procedures/Functions/Triggers)
2. Database Design and Normal Forms
3. Database Application Development
4. Storage Structures
5. Indexing and Hashing
6. Query Processing and Optimization
7. Transactions
8. Lock based Concurrency Control

Timelines

Course Start Date: 02 Jan 2023

Course End Date: 23 Mar 2023

Exam 1: 11th Feb 2023 (11:00 am - 12:30 pm)

Exam 2: 22nd or 23rd Mar 2023

Textbooks

Course Textbook

[Database Systems Concepts, A. Silberschatz, H. Korth and S. Sudarshan, McGraw Hill, 7th Edition](#)

Others

[Database Management Systems, R. Ramakrishnan and J. Gehrke, 3rd Edition](#)

[Fundamentals of Database Systems, R. Elmasri and S. B. Navathe, Addison Wesley, 7th Edition](#)

[Database Systems: The Complete Book, H. Garcia, J. Ullman, J. Widom, 3rd Edition](#)

Evaluation

Assignments (2-3): 50%

Exams (2): 50%

TAs

Suryamukhi K <cs17m19p100001@iith.ac.in>

Rahul Biswas <cs22resch01004@iith.ac.in>

Poonam Rajput <cs22resch01003@iith.ac.in>

Gajesh Sailesh Kumar <CS21MTECH11003@iith.ac.in>

AMIT KUMAR <CS21MTECH11007@iith.ac.in>

KISHAN NAYANBHAI BHINDE <CS21MTECH11016@iith.ac.in>

MEDHA RACHEL PANNA <CS22MTECH11003@iith.ac.in>

Akash K S <cs22mtech11012@iith.ac.in>

Rushikesh Kanubhai Vaishnav <cs22mtech11001@iith.ac.in>

Kashyap Parmar <ai21mtech14006@iith.ac.in>

Assignment 1 [Due on 29/01/2023]

Instructions

1. This assignment must be done individually.
2. Total marks for this assignment is 100.
3. 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.
4. If we find cases of copying, then all those who are involved will be given either FR grade or 0 marks for the assignment. Please don't share your code or report with anyone.

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

- b. Print the top-5 students with highest CGPA for departments: Psychology, Elec. Eng., and Civil Eng.
 - c. Print the top-5 students with highest CGPA for course ids: 237, 349, 735
- 4. Put the output.pdf in the rollnumber_firstname folder, zip it and upload in Google classroom.

Assignment 2 [Due on 20/02/2023]

Instructions

1. This assignment has to be done in groups of atmost 4.
2. Total marks for this assignment is 100.
3. There is no constraint on which tools or programming language you use. We are only interested in the end-product that you deliver. You should design your application in such a way that it is user friendly, has good performance, and is secure.
4. Please don't use any existing CQA software. You need to develop it yourself.
5. If we find cases of copying, then all those who are involved will be given either FR grade or 0 marks for the assignment. Please don't share your code with anyone.

Overview

In this assignment you will learn how to design a web based application that will allow users to interact with databases in a user-friendly manner. Your application will have a web based user interface at the frontend, and will have an interface with a database at the backend.

You will develop a Community Question Answer (CQA) website. You will use a CQA dataset from [StackExchange archive](#). Download the [Software Engineering](#) dataset. For converting the XML files into csv you may use [this](#) link. You can also see a sample SQL schema [here](#).

Some useful links

- [W3Schools](#)
- [Java Servlets](#) (Java)
- [Java Spring Framework](#)
- [Django Framework](#) (Python)
- [Django Tutorial](#)
- [Django Autocomplete](#)

Exercises

1. Create a database named cqadb. Create appropriate tables with constraints and then populate the tables using the above data dump.
2. **Login** - Create an account for each user with their default password being the same as their username. Users should be able to login to the system using username and password. Allow new users to create accounts. Use cookie to provide session to authenticated users. You can let the

cookie be stored permanently. Demonstrate that after the user removes the cookie, the user is logged out of the system.

3. **Autocompletion Search** - Support search of tag name and user display name using autocompletion. While showing the result, list both the name and id for tags and users.
4. **Search Posts** - Support the following three post search functionalities:
 - a. Search by user id - Given a user id list all the posts of the users. Allow sort by time, upvotes.
 - b. Search by single tag - Given a tag, list all the posts containing that tag. Allow sort by time, upvotes.
 - c. Search by multiple tags - Given more than one tag, list all the posts containing all the given tags. Allow sort by time, upvotes.
5. **Create Posts** - A user can create a post and assign tags to them. The tags have to be one of the existing tags only. It would be useful if the user can be helped with the tagging process through autocompletion.
6. **Answer to Posts** - A user can search for posts and give answers. The interface should show the posted question and all the answers.
7. **Edit Posts**: A logged in user should be able to see all the posts the user has written. The user should be able to update the content/tags of their own post or even delete them.