

# CSCI3170 Introduction to Database Systems (2013 Spring)

## Project – Course Enrollment System

<b>Group Registration Deadline:</b>	8 February 2013	23:59
<b>Phase 1 Deadline:</b>	8 March 2013	23:59
<b>Phase 2 Deadline:</b>	21 April 2013	23:59

---

### 1. Introduction

You are asked to implement a Course Enrollment System for a University to store the information about the students and courses. The system has to support interactive inquiries from the students in the University.

There are two phases. In phase 1, you are required to design the database for the system (including an ER-diagram and a relational schema which conforms to 3<sup>rd</sup> Normal Form). After the deadline of Phase 1, a suggested solution will be provided. You are required to use the suggested solution to complete Phase 2. In Phase 2, you are required to implement a Course Enrollment System (including a Java application and a web interface). Our tutors will give tutorials on how to connect to a database system and deploy your work on the required platform.

This is a group project and each group consists of three members. ONLY one copy of solution is required for each group. Please sign the group registration on our course homepage before the group registration deadline.

### 2. Milestones

#### Preparation

- Read the document thoroughly and make sure you can understand all the assumptions and regulations stated in Section 4.

#### Phase 1

- According to the data specifications in Section 3, design an ER-diagram and transform it into a relational schema which conforms to 3<sup>rd</sup> Normal Form.

#### Phase 2

- According to the suggested solution of Phase 1, write a Java application that fulfills all requirements stated in Section 5.
- Provide a web interface for the system that fulfills all requirements stated in Section 6.
- Test your system with different inputs.
- Write a user manual for your system.

### 3. Data Specifications

Data sets of the system are in ASCII file format. Your Java application is required to read the data set and insert the records into appropriate tables of the provided Oracle DBMS via JDBC API. There are four input files, a list of students, a list of courses, a list of course sections and a list of enrollment records.

#### 3.1. Students

Item Name	Format	Description
Student ID	Positive integer, 10 digits	The unique identifier of the student
Name	Non-empty string with at most 25 characters	The name of the student
Major	Non-empty string with at most 100 characters	The major of the student

#### 3.2. Courses

Item Name	Format	Description
Course code	8 characters	The unique identifier of the course
Name	Non-empty string with at most 100 characters	The name of the course
Credit	Positive integer, 1 digit	The number of credits for the completion of the course
Prerequisite	8 characters (for each course)	Course(s) that have to be completed before enrolling into the course

#### 3.3. Course Sections

Item Name	Format	Description
Course code	8 characters	The unique identifier of the course
Year	Positive integer, 4 digits	The academic year
Term	Positive integer, 1 digit	The term of the academic year
Quota	Positive integer, 2 digits	The maximum number of students allowed to enroll the course section
Instructor	Non-empty string with at most 25 characters	The name of the instructor
Lecture	3 characters (for each timeslot)	The timeslot(s) of the lecture(s), e.g. 'M1' ( <b>See Appendix I</b> )

#### 3.4. Enrollment Records

Item Name	Format	Description
Student ID	Positive integer, 10 digits	The unique identifier of the student
Course code	8 characters	The unique identifier of the course
Year	Positive integer, 4 digits	The academic year
Term	Positive integer, 1 digit	The term of the academic year
Grade	1 character	The course grade ( <b>See Appendix II</b> )

## 4. Assumptions and Regulations

### General

- The system is case sensitive.
- There is no duplicate row.
- All floating-point numbers are rounded to 2 decimal places.
- There is no NULL value in the tables of the database, unless stated otherwise.
- The current date is the current system date in the deployment machine.

### Students

- The student ID is unique.
- Some students may have the same name.
- A student can only have one major.

### Courses

- The course code is unique.
- Some courses may have the same name.
- The number of credits for a course is either 1, 2, or 3.
- A course can have none or any number of prerequisite courses.

### Course Sections

- A course can have at most one section in each term of an academic year.
- The term of an academic year is either 1, 2 or 3.
- The first term is from January to April.
- The second term is from May to August.
- The third term is from September to December.
- An instructor is uniquely identified by his/her name.
- Each section must have at least one lecture.

### Enrollment Records

- A student should take no more than 21 credits of courses in a term.
  - A student must pass all the prerequisite courses before taking a course.
  - A course cannot be enrolled if there is time collision.
  - A course cannot be enrolled if the course is full.
  - Grade D and above are regarded as pass and grade F is regarded as fail.
  - The grade of an enrollment record is NULL if the student has not completed the course.
  - Once the student passed a course, he/she cannot take the same course again.
  - Course Add/Drop cannot be performed if the term has ended.
- .

## 5. Requirements for the Java Application

You are required to write a simple command line application in Java. The application should consist of two interfaces. One is for administrator and the other is for students. For simplicity, users only need to provide their usernames for entering different interfaces (no password is required). The username of Administrator is “admin”. The usernames of students are their student IDs, which are stored in the database for authentication.

### 5.1. Administrator

- **Create all required tables in the database:** create all required tables in the database simultaneously based on the relational schema.
- **Delete all tables in the database:** delete all existing tables in the database simultaneously.
- **Load data into the database:** read the data from the input files and inserts the records into the appropriate tables in the database.
- **Show the information of the database:** for each existing table, show the table name and the number of records.
- **Show the ranking of courses:** Given an academic year and a term (user inputs), show the ranking numbers, course codes and names of courses that have a session on the specified term. A course has a high ranking if the course section has a high enrollment rate (enrollment total/quota) in the specified term. The courses are sorted by the ranking number in ascending order. If two courses have the same ranking, they will have the same ranking number and will be sorted by the course code in ascending order. Modified competition ranking is used in ordering the courses.

### 5.2. Student

- **Enroll into a course:** Given an academic year, a term and a course code (user inputs), enroll into the course session in the specified term.
- **Drop from a course:** Given an academic year, a term and a course code (user inputs), drop from the course session in the specified term.
- **Show the academic report:** Given an academic year and a term (user inputs), show the academic report in the specified term. The records are sorted by the course code in ascending orders and output as follows:

Course Code	Course Name	Credits	Grade	Points
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
Term GPA: ...				
Cumulative GPA: ...				

**P.S.**  $GPA = \text{SUM}(\text{Credits} * \text{Points}) / \text{SUM}(\text{Credits})$  (See Appendix II)

## 6. Requirements for the Web Interface

In addition to the command line application, you are required to provide a web interface for timetable planning. It is assumed that data were already inserted into the database by the JAVA application. The web interface should be as follows:

**Year & Term**  

2013 Term 1 ▼

**Course Code**  

Add

Drop

	Mon	Tue	Wed	Thu	Fri
08:30-09:15					
09:30-10:15		CSCI3180			
10:30-11:15				CSCI3170	
11:30-12:15			CSCI3170	CSCI3170	
12:30-13:15					
13:30-14:15					
14:30-15:15	CSCI3180				
15:30-16:15	CSCI3180				
16:30-17:15					
17:30-18:15					

**Course list**  
CSCI3170 Introduction to Database Systems (3 credits)  
CSCI3180 Principles of Programming Languages (3 credits)

- To use the timetable planner, users must first specify the academic year and term.
- The dropdown box “Year & Term” will consist of all terms that have at least one course for enrollment.
- Upon changing the options in the dropdown box “Year & Term”, the page will be reset.
- After specifying the academic year and term, users can add a course to the timetable by inputting the course code and clicking the button “add”.
- The added courses will be shown in the timetable and the course list immediately.
- Similarly, user can remove a course from the timetable and the course list by inputting the course code and clicking the button “drop”.
- Since the web interface does not require users to input their student IDs, prerequisite checking is not needed.

## 7. Grading Policy

- The marks are distributed as follows:

Phase	Content	Mark Distribution
1	ER-diagram	10%
	Relational schema (based on your ER-diagram)	5%
2	Java application	40%
	Web interface	40%
	User manual	5%

- No late submission is allowed.
- There will be a mark deduction if your application is terminated unexpectedly during the demonstration.
- There will be a mark deduction if you have to modify the source codes during the demonstration.
- All members in the same group will receive the same marks for the project. In order to encourage every student to participate in the project, a question about this project may be asked in the final examination.

## 8. Demonstration

You need to sign up for a demonstration to show your work for Phase 2. All group members should attend the demonstration. The duration for the demonstration for each group is about 20 minutes. We will test your Java application in a SPARC machine of our department. For the web interface, we will upload your source codes to our web server and test your work via Google Chrome. Please make sure that your program is compatible with the system settings. The data set used in the demonstration will be different from the data set provided for testing. The details about the demonstration will be posted on the course website.

## 9. Submission

### Phase 1

- Submit a PDF file (one copy for each group) to your Moodle account.

### Phase 2

- Submit a ZIP file (one copy for each group) to your Moodle account. The ZIP file should consist of all your source codes and the user manual (manual.pdf) for your system. For reference, please write a README file (README.txt), which contains:
  - Group information
  - List of files with description
  - Methods of compilation and execution

# Appendix

## I. Timeslots

	Mon	Tue	Wed	Thu	Fri
08:30-09:15	M1	T1	W1	H1	F1
09:30-10:15	M2	T2	W2	H2	F2
10:30-11:15	M3	T3	W3	H3	F3
11:30-12:15	M4	T4	W4	H4	F4
12:30-13:15	M5	T5	W5	H5	F5
13:30-14:15	M6	T6	W6	H6	F6
14:30-15:15	M7	T7	W7	H7	F7
15:30-16:15	M8	T8	W8	H8	F8
16:30-17:15	M9	T9	W9	H9	F9
17:30-18:15	M10	T10	W10	H10	F10

## II. Course grades

Grade	Points	Standard
A	4.0	Excellent
B	3.0	Good
C	2.0	Pass
D	1.0	
F	0.0	Failure