SUGANG – COURSE REGISTRATION WEB APPLICATION

AUTHOR: PHUOC KHANG BUI – 배복캉 (2021117446)

Department of Computer Science and Engineering, Kyungpook National University

I. INTRODUCTION:

A course registration system is necessary for every university to allow students to register for preferred courses in advance before a new semester starts. To achieve that goal, a Database Management System is required. This project was conducted in the Database class of the 2022 Fall Semester under Prof. Jung Inuk's instructions to demonstrate students' knowledge and ability to implement and maintain a DBMS.

***SCOPE:** This project only focuses on the relationship and interaction between Students and Courses.

***VERSION CONTROL:** *https://github.com/kangggchan/sugang_database2022*

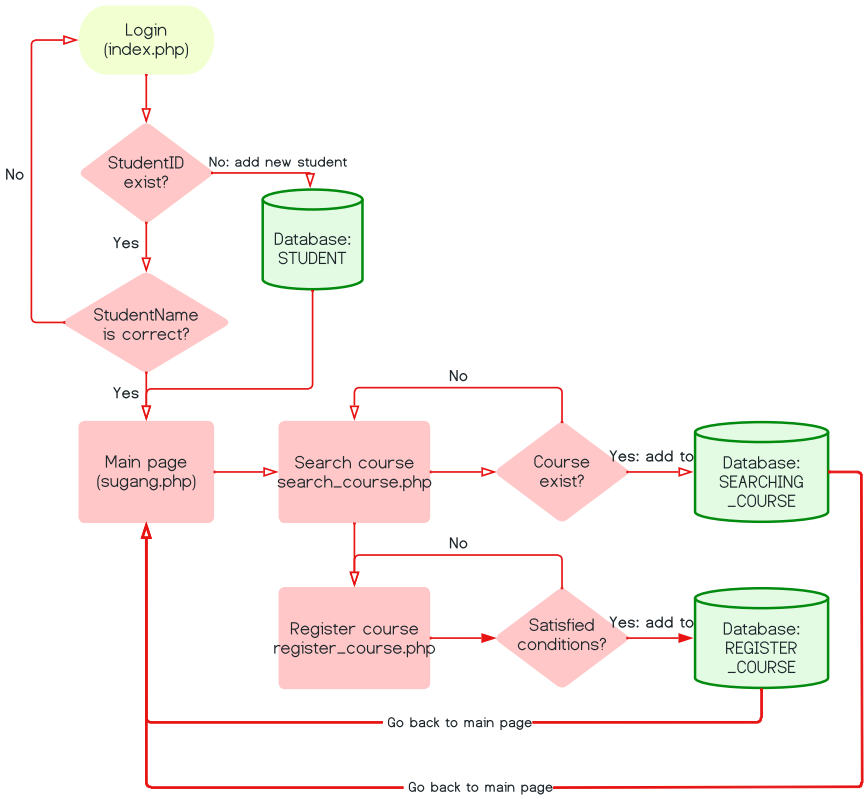
***DISCLAIMER:** The web UI package was collected from KNU's official *Sugang Website*.

II. FUNCTIONALITY-SPECIFIC REQUIREMENTS:

1. Shall have a login function to identify students who are accessing the web.
2. Shall display students' information on the main page.
3. Shall have a local Database Management System to store and maintain data.
4. Shall include README.md file to give an instruction on how to use and install.
5. Shall have a UI match with KNU's identities.
6. Shall allow students to search and register for courses with the following conditions:
    - Students can only register for courses designed for the 2022 Fall semester.
    - Each student can only register for up to 24 credits.
    - Each student can only register for up to 3 Liberal Arts courses.
    - Each student can only register for 1 class of each subject.
    - Each class has a quota of 2 students.
7. Shall allow students to delete registered courses.

Ⅲ. OPERATION:

1. PROCESS FLOW:



2. DATABASE MANAGEMENT SYSTEM:

2.1. Normalization:

- **First Normal Form (1NF):**

	.					1	CourseCode	..										.	RegisteredStudent	RegisteredStudentID
1	20…	2학기	1	교양	IT대학	컴…	CLTR0003-005	실용화법	3	3	0	김령환	화…	화	…	산…	251	70	배복캉	2021117446
2	20…	2학기	1	교양	IT대학	컴…	CLTR0003-005	실용화법	3	3	0	김령환	화…	화	…	산…	251	70	이재은	2020116999

As we can see, data is stored in one single table, and every record is unique, but we have a problem.

Assume that 20 students register for the course “실용화법”, so there will be 20 records for only one single course. That would take a lot of data storage, also making it difficult to manage and perform operations. So we need to switch to another form.

- **Second Normal Form (2NF):**

Now I split it into 2 separate tables, one for ***COURSES,*** one for ***STUDENTS.***

	...	.	.		..	...	Course...	Cour...		.		...					StudQuota
1	2022	2학기	1	교양	IT대학	컴퓨터학부...	CLTR0003-005	실용화법	3	3	0	김령환	화...	화...	산격...	251	70
2	2022	2학기	1	교양	IT대학	컴퓨터학부...	CLTR0003-006	실용화법	3	3	0	박소은	화...	화...	산격...	248	70
3	2022	2학기	2	교양	IT대학	컴퓨터학부...	CLTR0043-007	서양의역사와문화	3	3	0	최현미	화...	화...	산격...	B102	70
4	2022	2학기	3	교양	IT대학	컴퓨터학부...	CLTR0089-004	회계와사회생활	3	3	0	조성표	토...	토...	산격...	B102	105

The COURSE table is an entity with all of the courses' information and its attributes describing the properties of each course. The primary key for this table is **CourseCode**.

	StdName 1	StdNo	StdCollege	StdDepartment
1	배복캉	2021111111	IT대학	컴퓨터학부 글로벌소프트웨어융합전공
2	배복캉	2021117499	IT대학	컴퓨터학부 글로벌소프트웨어융합전공
3	도방영	2021117	IT대학	컴퓨터학부 글로벌소프트웨어융합전공

Similarly, the STUDENT table is an entity about students in school and its attributes describe the properties of each student. The primary key for this table is the **StdNo**.

But we are lacking something… The relation between those tables!

- **Third Normal Form plus (3NF+):**

Along with two tables from 2NF, I will add two more tables to store the student-course relation: COURSE_REGISTRATION and COURSE_SEARCHING.

	ReCourseNo	ReStdNo 1
1	COMP0216-001	2021117499
2	COME0331-008	2021117499
3	COME0311-005	2021117499
4	COME0301-014	2021117499
5	COME0301-011	2021117499

	SearchCourseNo	SearchStdNo
1	CLTR0003-005	2021117499
2	CLTR0003-006	2021117499
3	COMP0204-004	2021117499

Here, the structure of these two tables is similar to each other, they only differ in their uses.

COURSE_REGISTRATION: Stores the course registration information of students.

COURSE_SEARCHING: Stores the searching histories for displaying on the web.

The **StdNo** and **CourseCode** are Foreign Keys (FKs). And since the pairs between **StdNo** and **CourseCode** are unique, to avoid duplicate values (or one course is registered by a student multiple times in another words), I set those two attributes as a Composite Key in both two tables.

2.2. Implementation:

```sql
CREATE DATABASE SUGANG;
USE SUGANG;

CREATE TABLE COURSE (
  OpenYear INTEGER,
  Semester VARCHAR(3),
  StudYear INTEGER,
  Type VARCHAR(10) NOT NULL,
  College VARCHAR(30),
  Department VARCHAR(30),
  CourseCode VARCHAR(12) NOT NULL,
  CourseName VARCHAR(30) NOT NULL,
  Credit INTEGER,
  Lecture INTEGER,
  Practice INTEGER,
  Lecturer VARCHAR(10),
  CourseTime1 VARCHAR(255),
  CourseTime2 VARCHAR(255),
  LectureBuilding VARCHAR(30),
  LectureRoom VARCHAR(10),
  StudQuota INTEGER,

  Primary key (CourseCode),
  UNIQUE (CourseCode)
);

CREATE TABLE STUDENT (
    StdName VARCHAR(10),
    StdNo VARCHAR(10),
    StdCollege VARCHAR(30),
    StdDepartment VARCHAR(30),

    PRIMARY KEY (StdNo),
    UNIQUE (StdNo)
);

CREATE TABLE COURSE_REGISTRATION (
    ReCourseNo VARCHAR(14),
    ReStdNo VARCHAR(10),
    FOREIGN KEY (ReCourseNo) REFERENCES COURSE(CourseCode) ON UPDATE CASCADE,
    FOREIGN KEY (ReStdNo) REFERENCES STUDENT(StdNo) ON UPDATE CASCADE,
    PRIMARY KEY (ReStdNo, ReCourseNo)
);

CREATE TABLE COURSE_SEARCHING (
    SearchCourseNo VARCHAR(14) NOT NULL ,
    SearchStdNo VARCHAR(10) NOT NULL ,

    FOREIGN KEY (SearchCourseNo) REFERENCES COURSE(CourseCode) ON UPDATE CASCADE ,
    FOREIGN KEY (SearchStdNo) REFERENCES STUDENT(StdNo) ON UPDATE CASCADE,
    PRIMARY KEY (SearchCourseNo, SearchStdNo)

);
```

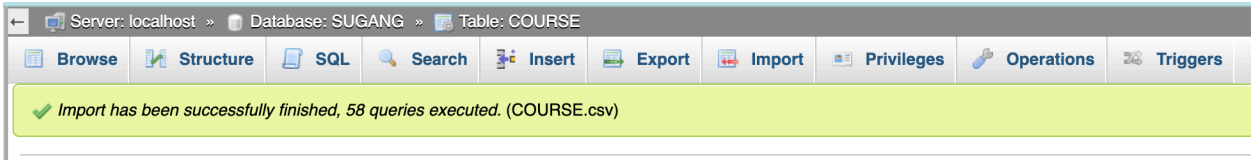
- **INSERTION:** I use phpMyAdmin to insert into table COURSE as a bulk, we can also use queries but I have some problems when using queries. phpMyAdmin seems to work more properly.

Using queries:

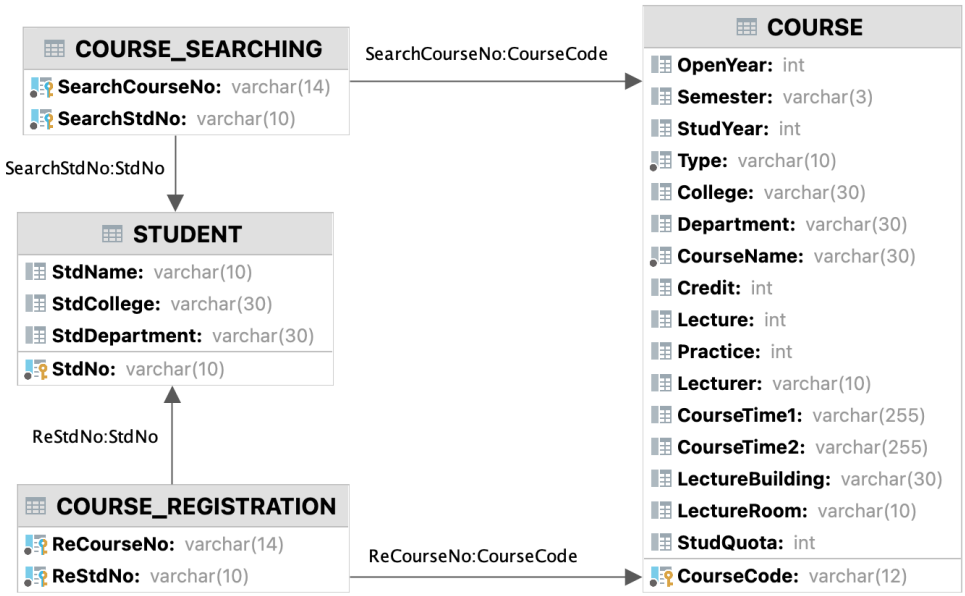
```
117
118  LOAD DATA INFILE '/Users/khangbuiphuoc/Sites/Sugang/data/COURSE.csv'
119      INTO TABLE COURSE
120      FIELDS TERMINATED BY ','
121      LINES TERMINATED BY '\n'
122      IGNORE 1 LINES;
```

[01000][1265] Data truncated for column 'StudQuota' at row 1

Using phpMyAdmin:



2.3. Entity Relationship Diagram (ERD):



2.4. Triggers:

To avoid students from registering for improper courses, I use several triggers in COURSE_REGISTRATION table:

LIBERAL_COURSE_LIMIT before insert — Avoid students from registering for more than 3 Liberal Arts courses.

```
CREATE TRIGGER LIBERAL_COURSE_LIMIT BEFORE INSERT ON COURSE_REGISTRATION
    FOR EACH ROW BEGIN
    IF (SELECT count(*) FROM COURSE_REGISTRATION WHERE ReCourseNo LIKE 'CLTR%'
                    AND ReStdNo = NEW.ReStdNo GROUP BY ReStdNo ) > 2
        AND NEW.ReCourseNo LIKE 'CLTR%' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'ERROR: Reach the limitation for liberal arts course!';
        end if;
END;
```

LIMIT_STD_IN_CLASS before insert — Avoid exceeding the number of students in a class.

```
CREATE TRIGGER LIMIT_STD_IN_CLASS BEFORE INSERT ON COURSE_REGISTRATION
    FOR EACH ROW BEGIN
    IF (SELECT count(*) FROM COURSE_REGISTRATION WHERE ReCourseNo = NEW.ReCourseNo) > 1 THEN
        SIGNAL SQLSTATE '45002'
        SET MESSAGE_TEXT = 'ERROR: Reach the limitation of student in class!';
    end if;
end;
```

TOTAL_COURSE_LIMIT before insert — Avoid students from registering for more than 24 credits.

```
CREATE TRIGGER TOTAL_COURSE_LIMIT BEFORE INSERT ON COURSE_REGISTRATION
    FOR EACH ROW BEGIN
    IF (SELECT sum(credit) FROM COURSE, COURSE_REGISTRATION
        WHERE CourseCode = ReCourseNo AND ReStdNo = NEW.ReStdNo)
           + (SELECT Credit FROM COURSE WHERE CourseCode = NEW.ReCourseNo) > 24 THEN
        SIGNAL SQLSTATE '45001'
        SET MESSAGE_TEXT = 'ERROR: Reach the limitation of credit!';
        end if;
END;
```

DUPLICATE_SUBJECT before insert — Avoid students from registering for the same subject.

```
CREATE TRIGGER DUPLICATE_SUBJECT BEFORE INSERT ON COURSE_REGISTRATION
    FOR EACH ROW BEGIN
    IF (SELECT CourseName FROM COURSE WHERE NEW.ReCourseNo = COURSE.CourseCode)
          IN (SELECT CourseName FROM COURSE WHERE CourseCode
              IN (SELECT ReCourseNo FROM COURSE_REGISTRATION WHERE NEW.ReStdNo = ReStdNo)) THEN
       SIGNAL SQLSTATE '45003'
       SET MESSAGE_TEXT = 'ERROR: You have already registered for that subject!';
end if;
end;
```

IV. ABOUT AUTHOR:

ABOUT ME: My name is ***Bui Phuoc Khang*** or people may know me as 배복캉. I am a sophomore student at KNU, Department of Computer Science and Engineering. I am a perfectionist and a curious student who loves to sit behind the computer and spend hours discovering things I don't know yet. Once the problem interests me, I would think about it all day long even while I'm sleeping. The next morning, when I woke up, is the time I find out the answer.

ABOUT THIS PROJECT: This is a project I'm very interested in because there were so many bugs that occurred while I was doing this project, and before that, I had no idea what *PHP* and HTML are. After spending hours doing research, reading PHP/MySQL manual documents, and surfing hundreds of question topics on StackOverflow, finally I made it. On this occasion, I want to give my finest thanks to Prof. Jung Inuk. Thanks to him, throughout his class, I not only learned how to establish a database but also learned about web development. Ultimately, now I can be confident to apply for a full-stack web developer position. Thank you for reading until the last words!