Table Structure

Database schema StudentEnrollment contains following tabels:

`class`: Repersents a class and includes information and has following columns |- class_id |- term |- course_number |- section_number |- class_description

IMPORTANT :In this table, term has to be entered specifically as below, otherwise the logic to get most recent class in ClassManagement will fail. - [ ] 'Fall2023' for Fall of 2023 - [ ] 'Spri2023' for Spring of 2023 - [ ] 'Summ2023' for Summer of 2023

`ctategory`: Defines different categories for gradeing within a class. This table has following columns. |- category_id |- weight |- category_name |- class_id

IMPORTANT : Weight has to be entered in decimal value. For example if category weight is 35% then entered value should 0.35.

`student`: Stores the information about students and has following columns. |- student_id |- username |- first_name |- last_name |- email_address |- contact_number |- address

`assignment`: Repersents assignments given in a class and has following columns. |- assignment_id |- assignment_name |- assignment_description |- point_value |- class_id |- category_id

`grades`: Stores grades of each students's assignment and has following columns. |- student_id |- assignment_id |- grade

`enrolled`: Tracks the enrollment of students in classes and has following columns. |- student_id |- class_id

Implementation

Limitation

1. If term is not entered in the format described above, this code will not work to get recent terms based on Terms and may fail in other functionalities also.
2. My command line argument input doesn't accept the spaces between any input, for example while adding an assignment, belowe argument will fail

- `add-assignment Homework 1 Assignments 1st homework 80` Rather entering spaces between `Homework 1` and `1st homework` please enter any special characters such as `-,_` for the code to work. You can modify above instuction as below:
- `add-assignment Homework-1 Assignments 1st_homework 80`

The `EnrollmentApp.java` is the main class which calls all the other class instances and takes the user inputs.

The `util.java` class contains all the methods such as *getCategoryId, checkStudentEnrolled, checkStudentExist, getCategoryId* etc. which helps to check the existence of students, categories assignments in database.

The `ActiveClass.java` class stores all the attributes of the current class and provides getter and setter methods of each attribute.

The `ClassManagement.java` class contains all the methods to create and manage the class. The functionalities of class management are *newClass* to the database if it doesn't already exist, *selectClass* with different parameters and *assignActiveClass* stores the result of selected class.

The `CategoryAssignment.java` class in Java provides methods for managing categories and assignments associated with a class. The functionalities of the class to add new categories to the database for a specific class if it doesn't already exist, based on the provided category name and weight. The `showAssignment` and `showCategories` methods display the categories and assignment of a given class by retrieving them from the database.

The `StudentManagement.java` class contains methods to manage the studnets. The class contains the functionalities to add students if not already exist in database and enroll studnets in the class, if students are not enrolled in the class. The `gradeAssignment` method assigned the grades to students for the assignment after verify that if student is enrolled in that course or not.

The `GradesManagement.java` class manages the grade record of students. The `studentGrades` displays student's current grade in each category and subtaotal of each category as well as student's overall grade in the class. The `gradebook` method displays the current class's gradebook as well as information of student along with their total grades in class

Execution

- On my Oynx profile, I am unable to upgrade Java from existing version 11. Therefore my this code is designed to run for Java 11 version. If you have different version, please ensure to have 11 version before execution of code.
- Also this code execution requires the MySql Java Driver added to the Oynx class path. You may please read class instruction on how to do that. I have reproduced these instructions as below as well. > Java DataBase Connectivity (JDBC) is an application programming interface for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. To use it with MySQL database, we need a connector that connects Java with MySQL. Such a connector is already installed on onyx in a java jar file at the following path:

  `/opt/mysql/mysql-connector-java-8.0.30.jar`

We need to add the jar to our CLASSPATH so that the Java runtime can find it. We can do this by adding the following command to the end of your ~/.bashrc file that is run when you login to onyx.

```
export CLASSPATH=/opt/mysql/mysql-connector-java-8.0.30.jar:.:$CLASSPATH
```

Make sure that when you add the above line to your .bashrc that it is all contained on one line. You can use kwrite ~/.bashrc to add the above line. Then you will need to login again for it to take effect.

Instruction to run the code

- The parameters for MySql connections are hardcoded in `EnrollmentApp.main` as below. If you are running the MySql on a different port or host, please ensure to change it before executing any of the code.

```
String hostName = "localhost";
int port = 49564;
String dbName = "StudentEnrollment";
String userName = "msandbox";
String password = "REPLACEME";
```

Below steps assume that you are running the code from Command Line. 1. Change your working directory to `/StudentManagement/src/` 2. Run the following command to compile the code. This will complie the code and create class binary files.

```
javac EnrollmentApp.java
```

4. Then you can execute below command to run the program

```
java EnrollmentApp
```