

Final Project

CPS3740: Database Management System

|  |  |
| --- | --- |
| Student Name: | Chun Chen |
| Student ID: | 1234930 |
| Lecture: | Dr. Hemn Barzan Abdalla |

Fall 2025

1. **Abstract**

This project built a simple system for students' university course scheduling and grade calculation. The system integrates the course scheduling and grade calculation systems and uses triggers to automatically record grade changes, ultimately enabling students to quickly manage their studies.

1. **Introduction**

**Background:** The lack of integration between the student course scheduling system, grade calculation system, and course syllabus creates inconvenience for students' course selection planning.

**Problem:** Students often need to switch between different systems to make a final decision, and this separation often hinders the decision-making process. For example, some courses require a prerequisite grade of >=2.4 or be in progress before selection is allowed, but the separation of the grading system and course selection system prevents users from quickly determining whether they meet the requirements.

This project primarily targets **students with course selection needs**, aiming to enable them to **quickly make decisions about course scheduling**. And also export GPA csv

1. **System Requirement**

* **Functional Requirements**

**Frontend:**

1. A rapid presentation of the courses in the four-year plan.
2. Quickly view prerequisite courses for selected courses.
3. Enter or modify course grades.
4. Record course grade entry or modification.
5. Unlock courses that have met prerequisite requirements.
6. Delete courses in courses table.

**Backend:**

1. Quick entry of course schedules
2. Quick entry of prerequisite course relationship tables
3. Two triggers for recording changes in course grades

* **Non-Functional Requirements**

1. Quick interaction with the UI
2. A simple and logically arranged UI
3. **System Design**

* **System Architecture**

**Python(Tkinter) for GUI:**

Class\_gpa\_management\_app.py

**SQLWorkbench for localhost:**

Class\_GPA\_Management\_SYS\_db. Sql

* **Database Design**

图示

AI 生成的内容可能不正确。

**Entity Relationship (ER) Modeling** **in Ternary relationship Class\_GPA\_Management\_SYS\_db**

**courses table:**The main table in the schema, with a non-identifying relationship with three other tables, the **course\_id** key is a primary key in the courses table, and other attributes in the courses key provide other relevant information about the course is described.

**course\_code:** The course’s Abbreviation in the four-year plan, like ‘ESL3035’.

**course\_name:** The course’s name(English description) in the four-year plan, like ‘ESL 0303 Academic oral presentation I’.

**credit:** The GPA credits scope between 0 to 9.

**category:** This time, only divide the category between major and required.

**Text:** Other word description for this course.

The **relationship** between the other tables:

**For the course\_prerequisites table:** One course can have one or many course\_id, and prereq\_course\_id in the prerequisite courses table, and each course\_id and prereq\_course\_id have relevant course\_id in the courses table.

**For grades\_current table:** One course can have one grades\_current record and one grades\_current record only related to one course.

**For grade\_changes table:** One course can have multiple GPA change records, and one change record can only have one course of information.

**grades\_current table:**

we have a primary key **grade\_id** and a Foreign key **course\_id**. The other attributes in this table represent the grade and credit for the course.

**grade\_point:** the GPA between (0.00 – 9.00).

**term:** the semester we take the course.

**status:** the type contains (normal/ retake/ correction/ in progress).

**comment:** the comment for this grade.

**updated\_at:** the time we add this grade.

**We also added two triggers to record the changes:**When inserting a new grade into the **grades\_current table**, the **Insert\_trigger** catches the change into **grade\_changes.**

When updating a grade in **grades\_current table,** the **Update\_trigger** catches the change into **grade\_changes**.

The difference between the two triggers is that we do not record **old\_grade\_point** into **grade\_changes** **table** because we do not have old GPA before.

**grade\_changes table:**

This table records the grade change for each course from the trigger. The primary key change\_id, with Foreign key course\_id.

**old\_grade\_point:** the old GPA before(if no GPA before enter ‘NULL’).

**new\_grade\_point:** the new GPA when we Normal/retake/correction/In progress the class.

**change\_type:** the four statuses: Normal/retake/correction/In progress for the course.

**change\_time:** the time we record the change.

**course\_prerequisites table:**

This table records between the course and the prerequisite course, the primary key is ‘id’, and the two foreign keys are ‘course\_id’ and ‘prereq\_course\_id’, which reference to courses table.

**course\_prereq\_mapping table:**

This table helps us have a quick correspondence between the course and the prerequisite course.

* **GUI Design**

The GUI design contains three parts

| ----- UI Construction (Construct the Arrange the specific images and button layout of the user interface. )

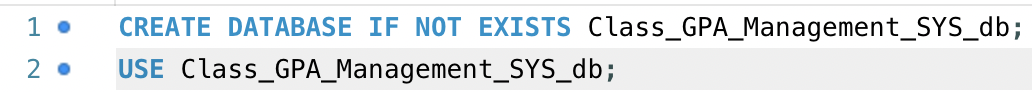
| ----- Event handlers (Define the programs that different buttons in the UI can trigger.)

| ----- Refresh functions (Define the specific statements of the program required for the response button.)

1. **Implementation**

* **Class\_GPA\_Management\_SYS\_db. Sql**

First, we create and use the schema:



Second, we create five tables:







with two triggers:

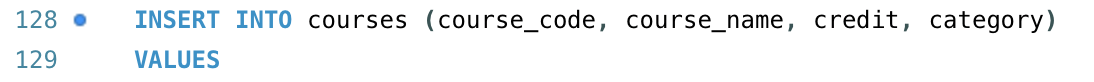
文本

AI 生成的内容可能不正确。

文本

AI 生成的内容可能不正确。

Then, we insert values into the courses and course\_prereq\_mapping table to **construct a basic course framework**





Finally, we select all the tables attributes

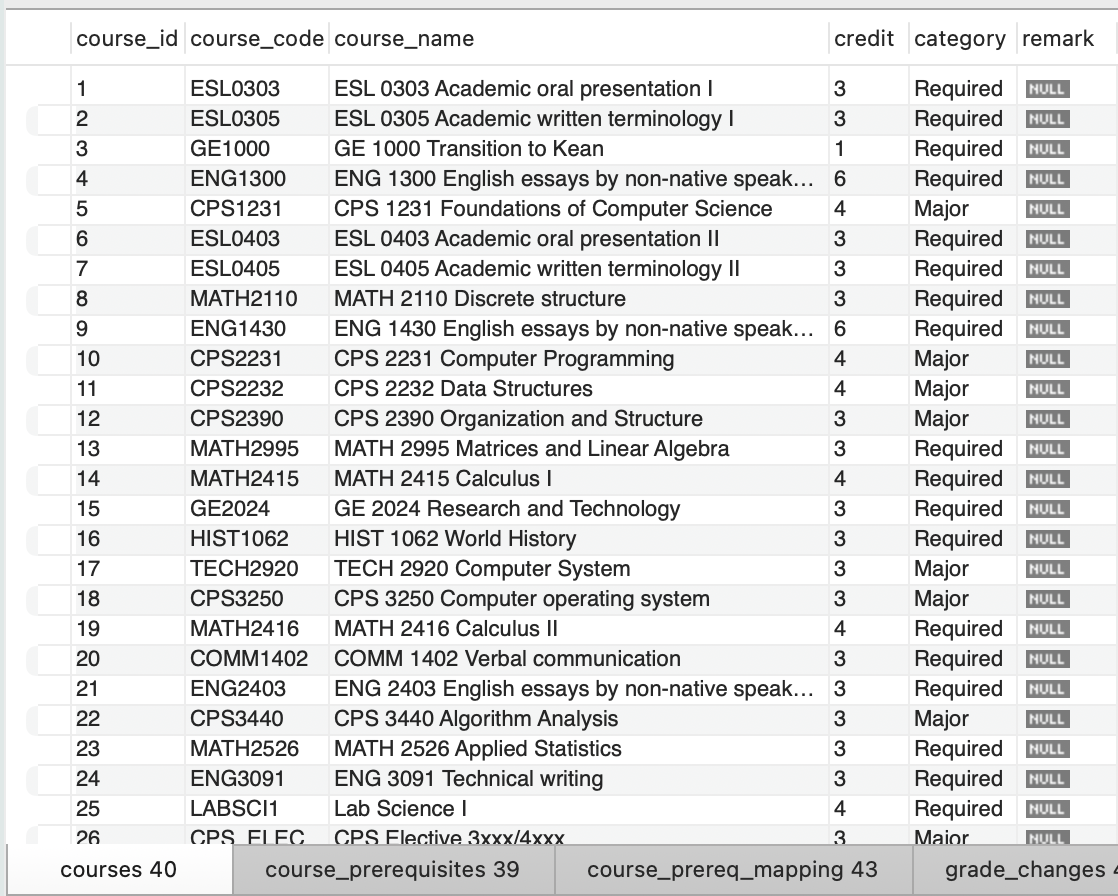
表格

AI 生成的内容可能不正确。

表格

AI 生成的内容可能不正确。表格

AI 生成的内容可能不正确。



The grades\_current and grade\_changes tables no value available at the moment.

* **Class\_gpa\_management\_app. Py**

In the program, we defined a function to let other functions have a quick connection with the database.

图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。

**The most basic operating logic is:**

1. we defined several functions to get a connection with the database and write SQL code to deal with different Event handlers.

|  
|  
|  
|  
|  
|

\/

1. Different methods are applied to handle different refresh requests.

|  
|  
|  
|  
|  
\/

1. Different update requests were used to process different entries and UI commands.

For example, we define the fetch\_unlocked\_courses function to connect with the database and **tuple** the courses that satisfy the prerequisite courses’ GPA >= 2.4 or the status = In progress, and the course grade must is NULL.

图片包含 文本

AI 生成的内容可能不正确。

And in one refresh function we use tuple return from fetch\_unlocked\_courses function to show the content.

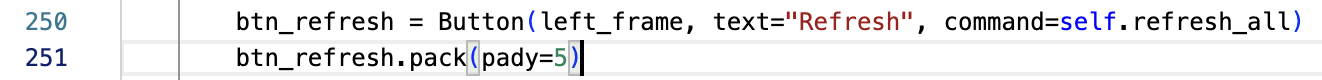
图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

手机屏幕截图

AI 生成的内容可能不正确。

Here is the command that we call the refresh command.



**Accessibility Features:**

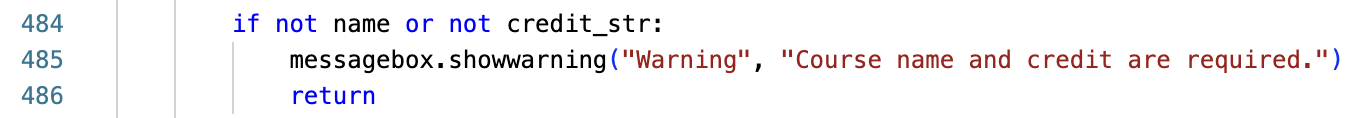
In SQL we define functions to insert course information and prerequisite courses information individually.

Also, some functions to drop values in the schema.

图片包含 文本

AI 生成的内容可能不正确。

In the GUI, we also built some functions to deal with some illegal requests.



1. **Testing**

We first start the workbench, then we run the code in the terminal:



Then we get the GUI with courses and grades\_current list:

**图形用户界面, 应用程序

AI 生成的内容可能不正确。**

If we select one course, we will get the prerequisites and grade change history for the selected course.

**图形用户界面, 应用程序

AI 生成的内容可能不正确。**

After we enter the grade into the GUI we save the data into grades\_current and grade\_changes(by trigger),

图形用户界面, 应用程序

AI 生成的内容可能不正确。

表格

AI 生成的内容可能不正确。表格

AI 生成的内容可能不正确。

Because the refresh function, the unlocked and grade change history list also refreshed.

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

1. **Conclusion**

This project initially integrates a four-year plan, a grading system, and a course scheduling system, enabling rapid entry of grades and course planning. However, some course selection logic is still incomplete and requires further improvement.