Student Elective Allocation System

Project Overview

The **Student Elective Allocation System** is designed to automate the allocation of elective subjects to students based on their **preferences** and **GPA**. The system ensures that students are assigned subjects they prefer, while also considering subject availability, maintaining fairness in subject distribution, and ensuring that every student gets an elective whenever possible.

Table of Contents

Introduction

The allocation process is based on the **GPA** of students and their personal **subject preferences**. The students are prioritized based on their GPA, with higher GPA students getting the first priority for their preferred electives. This ensures a more meritocratic allocation system, while still taking into account students' choices.

Key Features:

- **Priority Allocation**: Students with a higher GPA are given priority for their top-choice electives.
- **Fair Distribution**: The system ensures that no subject gets overbooked while ensuring all students are allotted electives.
- **Dynamic Adjustment**: If students' first-choice electives are unavailable, the system allocates their second-choice elective, and so on.
- **Exception Handling**: If no elective can be allocated to a student, they are stored in the **UnallotedStudents** table for further review.

Database Schema

The system utilizes five key tables, each of which serves a distinct purpose. Below is a description of each table:

1. StudentDetails

This table contains basic information about the students.
Columns:

- **StudentID**: Unique identifier for each student.
- **Name**: Student's name.
- **GPA**: Grade Point Average of the student.
- **Year**: The year the student is in.

2. SubjectDetails

This table contains the list of elective subjects available to students.
Columns:

- **SubjectID**: Unique identifier for each subject.
- **SubjectName**: Name of the subject.
- **AvailableSeats**: Number of seats available in the subject.

3. StudentPreference

This table contains students' preferences for elective subjects, where students rank their preferred subjects.
Columns:

- **StudentID**: The ID of the student.
- **SubjectID**: The ID of the subject.
- **PreferenceRank**: A rank (1 for the top preference, 2 for the second, etc.).

4. Allotments

This table stores the subjects allocated to each student.
Columns:

- **StudentID**: The ID of the student.
- **SubjectID**: The ID of the subject allocated to the student.

5. UnallotedStudents

This table contains students who were not allocated any elective subjects.
Columns:

- **StudentID**: The ID of the student.
- **Reason**: The reason why the student couldn't be allocated an elective (e.g., no subjects available).

Setup Instructions

To set up and run the Student Elective Allocation System, follow the steps below:

1. Create and Populate Required Tables

Before running the allocation procedure, you need to set up the necessary database tables and populate them with sample data.

To get started, download the SQL script to create and populate the tables from the following GitHub link:
https://github.com/ChetanyaYadav/Celebal-Internship

2. Create the Stored Procedure

The core of this system is a stored procedure that allocates the elective subjects based on student preferences and GPA. You need to create the procedure in your SQL database by executing the following code:

The stored procedure should contain logic for:

1. Sorting students by GPA.
2. Allocating subjects based on student preferences, considering available seats.
3. Updating the **Allotments** table for successful allocations.
4. Storing unallocated students in the **UnallotedStudents** table.

Running the Allocation Procedure

Once you've set up the tables and created the stored procedure, you can run the procedure to allocate electives to the students.

Verification

After running the allocation procedure, you can verify the results by querying the relevant tables:

-- To check the students who were not allocated any subjects

You should see:

- A list of students along with the allocated elective subjects in the **Allotments** table.
- Students who couldn't be allocated any subjects (if any) in the **UnallotedStudents** table.

Procedure Breakdown

Here’s a step-by-step breakdown of how the allocation procedure works:

1. **Fetch Students and Sort by GPA**:
   The first step is to retrieve all students from the **StudentDetails** table and sort them by GPA in descending order.
2. **Iterate Over Each Student**:
   For each student, check their subject preferences (from the **StudentPreference** table). Attempt to allocate the top-ranked subject first.
3. **Allocate Subjects**:
   For each subject, check if there are available seats in the **SubjectDetails** table. If seats are available, allocate the subject to the student and reduce the seat count.
4. **Handle Unallocated Students**:
   If no elective is available for a student (i.e., all preferred subjects are full), record the student in the **UnallotedStudents** table.

Performance Considerations

In large-scale systems, performance is crucial. Consider the following performance improvements:

- **Indexing**: Add indexes on columns like StudentID and SubjectID in the relevant tables to improve query performance.
- **Batch Processing**: If the system has a very large number of students, break the allocation process into smaller batches to avoid timeouts and reduce system load.
- **Concurrency**: Ensure that the procedure can handle multiple concurrent executions without conflicts, especially when updating the seat count for subjects.

Contact

For any questions, concerns, or suggestions, feel free to reach out to me:

- **Name**: Chetanya Yadav
- **Email**: yadavchetanya111@gmail.com
- **GitHub**: https://github.com/ChetanyaYadav
- **LinkedIn**: https://www.linkedin.com/in/chetanya-yadav-07a048207/

Appendix:

If you need any additional information or resources, please don't hesitate to reach out to me directly. I'll be happy to assist with any aspect of the system, whether it's related to SQL, system design, or optimization techniques.