Course Name: Database Systems Laboratory

Course Tittle: CSE 3110

Name: Dipesh Talukdar

Roll:2007063

Dept: Computer Science and Engineering

University: Khulna University of Engineering and Technology

**Introduction**

**➢ Project Overview**

“Semester Planning & Student Engagement System” is a comprehensive system designed to streamline the management of events, workshops, and semester planning within educational institutions. Our platform offers seamless integration of course planning, student registration, and feedback mechanisms, ensuring a smooth and efficient experience for both organizers and participants.

**➢ Importance of the Database in this Project**

In your project, a well-designed and maintained database plays a pivotal role in ensuring the smooth operation and efficiency of various aspects. Here are some key points highlighting the importance of a database in your project:

1. **Data Organization**: The database serves as a central repository for organizing vast amounts of data related to events, workshops, semester plans, student registrations, feedback, and more. Proper organization facilitates easy retrieval and manipulation of data, enabling efficient decision-making and analysis.
2. **Data Integrity**: By enforcing constraints, such as primary keys, foreign keys, and data validation rules, the database ensures data integrity. This helps prevent inconsistencies, inaccuracies within the system, maintaining the reliability and trustworthiness of the information stored.
3. **Efficient Querying**: With an efficient database design and indexing strategies, querying operations can be optimized for performance. This enables quick retrieval of relevant information, supporting various functionalities such as event registration, workshop enrollment, and feedback analysis in real-time or near real-time.
4. **Integration and Interoperability**: A well-designed database facilitates seamless integration with other systems and applications, enabling data exchange and interoperability. This allows for the integration of third-party tools, APIs, or analytics platforms, enhancing the functionality and value proposition of the system.

Project Objectives

The primary objectives of “Semester Planning & Student Engagement System “ are to provide a comprehensive solution for the efficient management of events, workshops, semester planning, and student engagement within educational institutions. The project aims to achieve the following specific goals:

1. **Event Management**: Develop a system that enables organizers to effortlessly create, schedule, and manage various events, including conferences, seminars, and cultural programs. Facilitate online registration, payment processing, and attendance tracking to ensure smooth event execution.
2. **Efficient Workshop Coordination**: Implement functionalities to coordinate workshops on diverse topics, catering to the academic and professional development needs of students. Enable seamless registration, fee collection, and resource allocation for workshop sessions conducted within the institution.
3. **Optimized Semester Planning**: Design tools to assist academic departments in planning and scheduling courses for different semesters. Ensure compatibility with institutional guidelines and regulations, allowing for efficient allocation of faculty resources and classroom facilities.
4. **Student Registration Management**: Develop a robust system for student registration in courses, events, and workshops offered by the institution. Enable students to browse available offerings, register for their desired selections, and receive confirmation of enrollment status in real-time.
5. **Feedback Collection and Analysis**: Incorporate mechanisms for collecting feedback from students regarding courses, events, and workshops attended. Provide features for students to submit comments, ratings, and suggestions, and facilitate the analysis of feedback data to identify areas for improvement and enhancement.

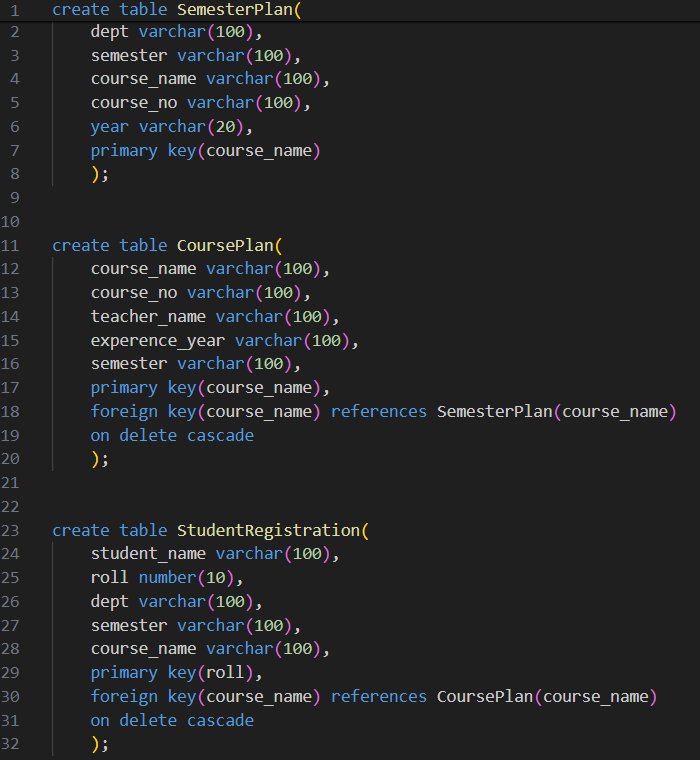
DatabaseDesign

**➢ Overview of the Database Schema**

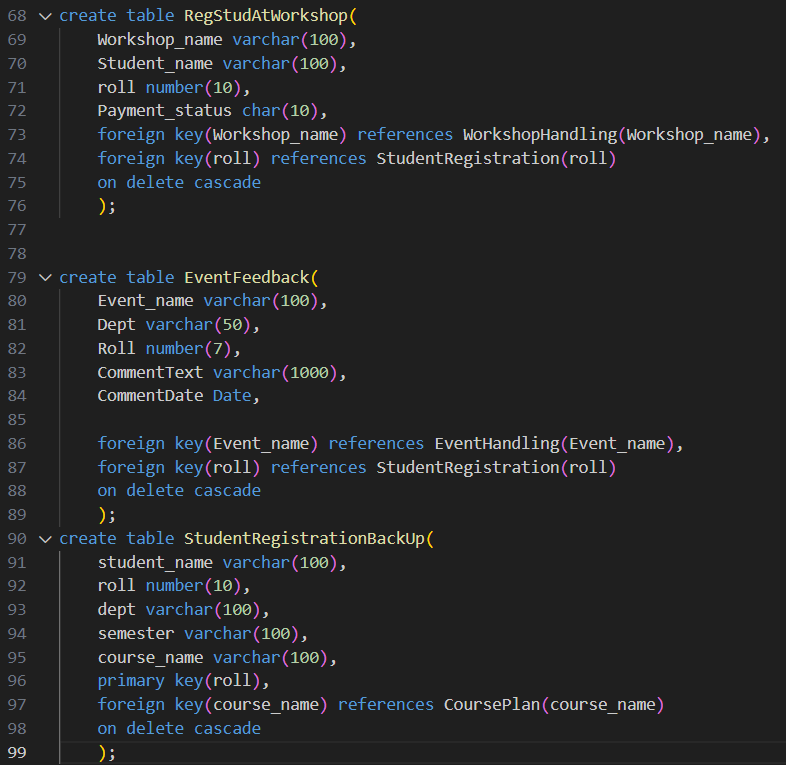
1. **SemesterPlan**:
   * **Description**: The SemesterPlan table stores information about the academic semesters offered by the institution, along with the courses available in each semester.
   * **Attributes**:
     + dept: The department offering the course.
     + semester: The academic semester.
     + course\_name: The name of the course (primary key).
     + course\_no: The course number.
     + year: The academic year in which the course is offered.
2. **CoursePlan**:
   * **Description**: The CoursePlan table contains details about the courses offered by the institution, including course instructors, semester details, and departmental affiliations.
   * **Attributes**:
     + course\_name: The name of the course (primary key).
     + course\_no: The course number.
     + teacher\_name: The instructor's name.
     + experience\_year: The instructor's years of experience.
     + semester: The academic semester in which the course is offered.
     + dept: The department offering the course.
3. **StudentRegistration**:
   * **Description**: The StudentRegistration table records students' registrations for courses offered by the institution, linking students to specific courses and academic semesters.
   * **Attributes**:
     + student\_name: The name of the student.
     + roll: The student's roll number (primary key).
     + dept: The department in which the student is enrolled.
     + semester: The academic semester in which the student is registered.
     + course\_name: The name of the course the student is registered for.
4. **StudentRegistrationBackUp**:
   * **Description**: The StudentRegistrationBackUp table serves as a backup for student registration data, providing redundancy in case of data loss or corruption.
   * **Attributes**:
     + student\_name: The name of the student.
     + roll: The student's roll number (primary key).
     + dept: The department in which the student is enrolled.
     + semester: The academic semester in which the student is registered.
     + course\_name: The name of the course the student is registered for.
5. **EventHandling**:
   * **Description**: The EventHandling table stores information about various events organized by the institution, including event names, registration fees, venues, types, and dates.
   * **Attributes**:
     + Event\_name: The name of the event (primary key).
     + Registration\_Fees: The registration fees for the event.
     + venue: The location where the event will be held.
     + Event\_Type: The type or category of the event.
     + Event\_Date: The date of the event.
6. **RegStudAtEvent**:
   * **Description**: The RegStudAtEvent table maintains records of students registered for events, tracking their attendance and payment status.
   * **Attributes**:
     + Event\_name: The name of the event.
     + Student\_name: The name of the registered student.
     + roll: The student's roll number.
     + Payment\_status: The payment status of the student for the event.
7. **EventFeedback**:
   * **Description**: The EventFeedback table captures feedback provided by students regarding events they have attended, including comments, dates, and departmental affiliations.
   * **Attributes**:
     + Event\_name: The name of the event.
     + Dept: The department of the student providing feedback.
     + Roll: The roll number of the student providing feedback.
     + CommentText: The feedback comments provided by the student.
     + CommentDate: The date the feedback was submitted.
8. **WorkshopHandling**:
   * **Description**: The WorkshopHandling table stores details about workshops conducted by the institution, including workshop names, registration fees, venues, types, and dates.
   * **Attributes**:
     + Workshop\_name: The name of the workshop (primary key).
     + Registration\_Fees: The registration fees for the workshop.
     + venue: The location where the workshop will be held.
     + Workshop\_Type: The type or category of the workshop.
     + Workshop\_Date: The date of the workshop.
9. **RegStudAtWorkshop**:
   * **Description**: The RegStudAtWorkshop table maintains records of students registered for workshops, tracking their attendance and payment status.
   * **Attributes**:
     + Workshop\_name: The name of the workshop.
     + Student\_name: The name of the registered student.
     + roll: The student's roll number.
     + Payment\_status: The payment status of the student for the workshop.

**➢ Table Relationships**

The relationships between the tables are primarily defined by foreign keys that ensure data integrity and relational connections

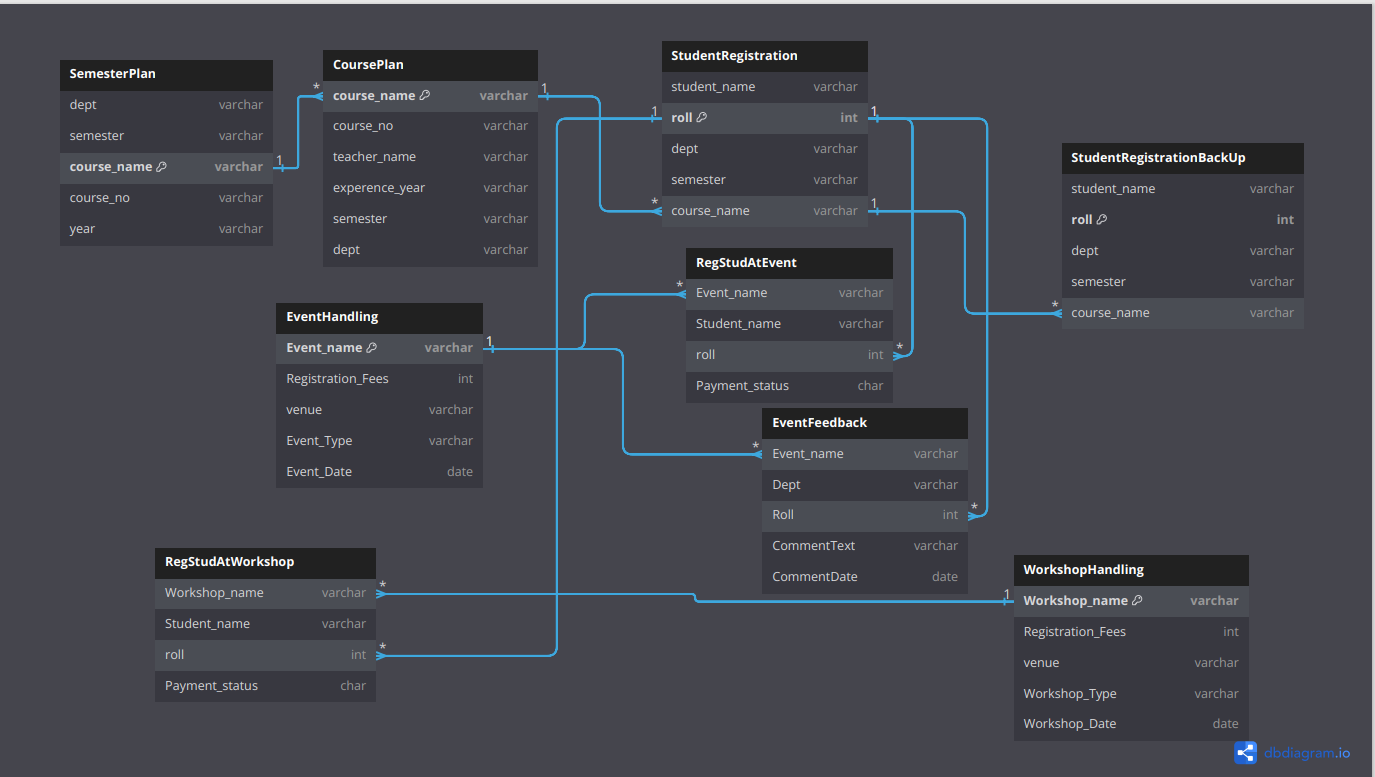
****





**➢ Entity-Relationship Diagram (ERD)**

The ERD diagram for this project illustrates the relationships between various entities and their attributes within the database. It visually represents how different tables are connected and interact with each other. Here's an overview of the key components of the ERD diagram:

****

**Fig: ERD Diagram of “Semester Planning & Student Engagement System” Project**

**SQL Queries and Functionality**

**➢ Examples of SQL Queries**

To facilitate a wide range of informational needs for users, several SQL queries have been designed to extract useful information from the database. Here are examples illustrating common queries:

• Retrieving Player Statistics:

Select Student\_name,roll from RegStudAtEvent where event\_name='Seminar on AI';

this query retrives the student information who registered in a specific event.

• Event Halndling Information

SELECT event\_name, venue, event\_type from EventHandling where event\_date = TO\_DATE('2024-11-15', 'YYYY-MM-DD');

this query retrives the event information happened in a specific date.

**➢ Queries for Common Operations**

In addition to specific queries, the database supports various common operations:

• To count the total number of students registered in each events:

SELECT Event\_name, COUNT(\*) AS total\_students\_registered From RegStudAtEvent GROUP BY Event\_name;

this query will show the total number of students registered in each event.

• To Retrive the Event and workshop Organized on a same date:

SELECT

e.Event\_name AS Event\_Name, e.Event\_Date AS Event\_Date,

w.Workshop\_name AS Workshop\_Name, w.Workshop\_Date AS Workshop\_Date

FROM

EventHandling e

JOIN

WorkshopHandling w ON e.Event\_Date = w.Workshop\_Date;

**➢ Implementation of Advanced SQL Features**

Advanced SQL features such as stored procedures, triggers, and functions are extensively utilized to enhance the functionality and efficiency of the database:

Triggers: When the registration fees of a workshop will be updated to 0,then the payment status of the students who registered for that event will be updated to 'Free'.

set serveroutput on

CREATE OR REPLACE TRIGGER WorkshopRegistrationUpdate

AFTER UPDATE OF Registration\_Fees ON WorkshopHandling

for each row

begin

IF :NEW.Registration\_Fees = 0 then

update RegStudAtWorkshop

set Payment\_status = 'Free'

where Workshop\_name = :NEW.Workshop\_name;

end IF;

end;

/

-- Simulate an update operation on EventHandling table

update WorkshopHandling

set Registration\_Fees = 0

where Workshop\_name = 'Data Science Workshop';

select \* from RegStudAtWorkshop;

Procedures: Used for complex operations like, a procedure that updates the payment status of a student for a particular event or workshop.

CREATE OR REPLACE PROCEDURE proc2(

event\_workshop\_name\_in IN VARCHAR,

student\_name\_in IN VARCHAR,

new\_status\_in IN CHAR

)

AS

BEGIN

UPDATE RegStudAtEvent

SET Payment\_status = new\_status\_in

WHERE Event\_name = event\_workshop\_name\_in AND Student\_name = student\_name\_in;

UPDATE RegStudAtWorkshop

SET Payment\_status = new\_status\_in

WHERE Workshop\_name = event\_workshop\_name\_in AND Student\_name = student\_name\_in;

END;

/

set serveroutput on;

BEGIN

proc2('Seminar on AI', 'Tina', 'Paid');

END;

/

select \* from RegStudAtEvent;

**Conclusion**

In the journey of developing “Semester Planning & Student Engagement System “ our aim was to create a comprehensive solution that addresses the complex challenges faced by educational institutions in managing events, workshops, semester planning, and student engagement. Throughout the development process, our focus remained on crafting a system that not only streamlines administrative tasks but also enhances the overall learning experience for students.

One of the key strengths of this project lies in its ability to harness data for informed decision-making and continuous improvement. Through feedback mechanisms and analytics tools, educational institutions can gain valuable insights into student preferences, program effectiveness, and areas for enhancement. This data-driven approach fosters a culture of adaptability and innovation, ensuring that institutions remain responsive to the evolving needs. In conclusion, this project represents more than just a software solution—it embodies a commitment to excellence, innovation, and inclusivity.