འབྲུག་རྒྱལ་འཛིན་གཙུག་ལག་སློབ་སྡེ།

SAMTSE COLLEGE OF EDUCATION

Mini Project on Club Management System in Samtse College of Education

Jamyang Choden (08230128)
Sangay Choden (08230183)
Tashi Yangzom (08230213)
Tenzin Choden (08230214)
Karma Wangchuk (08230136)

Samtse College of Education
Samtse
Royal University of Bhutan

Database Design and Development (DDD201)
Mrs. Sapna Thapa
Date: 29/05/2025

༄། །འབྲུག་རྒྱལ་འཛིན་གཙུག་ལག་སློབ་སྡེ།
བསམ་རྩེ་ཤེས་རིག་མཐོ་རིམ་སློབ་གྲྭ།

Royal University of Bhutan
Samtse College of Education

P.O. Box No. 329, Samtse, Bhutan

PLAGIARISM DECLARATION FORM

This form must be completed, signed and appended to each assignment you submit for marking in any form (Print or electronically).

Student Name: Jamyang Choden, Sangay Choden, Tashi Yangzom, Tenzin Choden, Karma Wangchuk
Student Number: 08230128, 08230183, 08230213, 08230214, 08230136
Module No. & Title: DDD201 Database Design and Development
Assignment No. & Title: Mini Project on Club Management System in Samtse College of Education
Section H2 of the Royal University of Bhutan's *Wheel of Academic Law* provides the following definition of academic dishonesty:
"Academic dishonesty may be defined as any attempt by a student to gain an unfair advantage in any assessment. It may be demonstrated by one of the following:

- **Collusion**: the representation of a piece of unauthorized group work as the work of a single candidate.
- **Commissioning**: submitting an assignment done by another person as the student's own work.
- **Duplication**: the inclusion in coursework of material identical or substantially similar to material which has already been submitted for any other assessment within the University.
- **False declaration**: making a false declaration in order to receive special consideration by an Examination Board or to obtain extensions to deadlines or exemption from work.
- **Falsification of data**: presentation of data in laboratory reports, projects, etc., based on work purported to have been carried out by the student, which have been invented, altered or copied by the student.
- **Plagiarism**: the unacknowledged use of another's work as if it were one's own.

Examples are:

- verbatim copying of another's work without acknowledgement
- paraphrasing of another's work by simply changing a few words or altering the order of presentation, without acknowledgement
- ideas or intellectual data in any form presented as one's own without acknowledging the source(s)
- making significant use of unattributed digital images such as graphs, tables, photographs, etc. taken from test books, articles, films, plays, handouts, internet, or any other source, whether published or unpublished
- submission of a piece of work which has previously been assessed for a different award or module or at a different institution as if it were new work
- use of any material without prior permission of copyright from appropriate authority or owner of the materials used"

Student Declaration

I confirm that I have read and understood the above definitions of academic dishonesty. I declare that I have not committed any academic dishonesty when completing the attached piece of work.

Student's Signature: 　　　　　　　　　　　　**Date of signing**:

Club Management System – Project Report

1. Introduction

The Club Management System is designed to streamline and automate the management of club-related activities within a college setting. It facilitates smooth registration and tracking of members, monitors attendance, and generates participation certificates. The system enhances communication within clubs through integrated notifications and announcements. With a user-friendly interface, it is accessible via both mobile devices and computers, ensuring ease of use for all users.

2. Discussion

The development of the Club Management System was driven by the need to address key issues such as inefficient record-keeping, poor communication, and lack of recognition for student participation. The mixed-methods approach provided valuable insights into student expectations and challenges, which helped in shaping a user-centered system design.

Survey data revealed that while many students were interested in club activities, barriers such as difficult registration processes and inconsistent communication discouraged continued involvement. Interviews and observations confirmed that students often felt disconnected from club operations, especially when attendance or contributions went unrecognized.

By integrating features like automated certificate generation, attendance tracking, and a notification system, the new system directly addresses these concerns. Document analysis also highlighted gaps in past participation records, reinforcing the need for a centralized, digitally accessible solution. The design process, although challenged by limited time and technical constraints, was informed by continuous feedback and iterative development.

Overall, the Club Management System not only streamlines club operations but also empowers students by ensuring their efforts are recorded, recognized, and rewarded. This project highlights how thoughtful use of technology can foster a more engaging and organized extracurricular environment in educational institutions.

3. Scope of the project

The club management system is designed in such a way that it helps to manage all the club-related activities in an organized and productive manner. It allows easy registration and management of the members which includes tracking their attendance (maintaining good attendance) and generating certificates for those members who complete the required period of participation. Communication within the club is made easier through notifications and announcements. The system has a user-friendly interface, making it easy to use both on mobile and computers.

4. Problem Statement

The college’s current approach to managing student club faces several key challenges**:**

The current club enrollment process, which involves filling out an Excel sheet provided by the Dean of Student Affairs (DSA) and attending mandatory orientation sessions, is often seen as ineffective due to the lack of active engagement during orientations. Additionally, the credibility of membership certificates in some clubs is undermined by poor participation tracking, allowing students with minimal involvement—or those who repeatedly enroll solely for certification—to receive the same recognition as consistently active members, which discourages genuine commitment. Moreover, the absence of a centralized monitoring system makes it difficult for the college to track overall student participation in extracurricular activities, limiting its ability to identify disengaged students and to develop targeted strategies to promote more meaningful student involvement in club activities.

5. Feasibility of the project

The proposed Club Registration System is highly feasible from multiple perspectives. Technically, the system can be developed using widely available technologies such as web development tools and relational databases. These technologies are well-supported and require no specialized hardware or software, making the system technically viable. Moreover, the skills required for development—such as frontend and backend programming—are commonly possessed by software developers or can be easily acquired through short-term training.

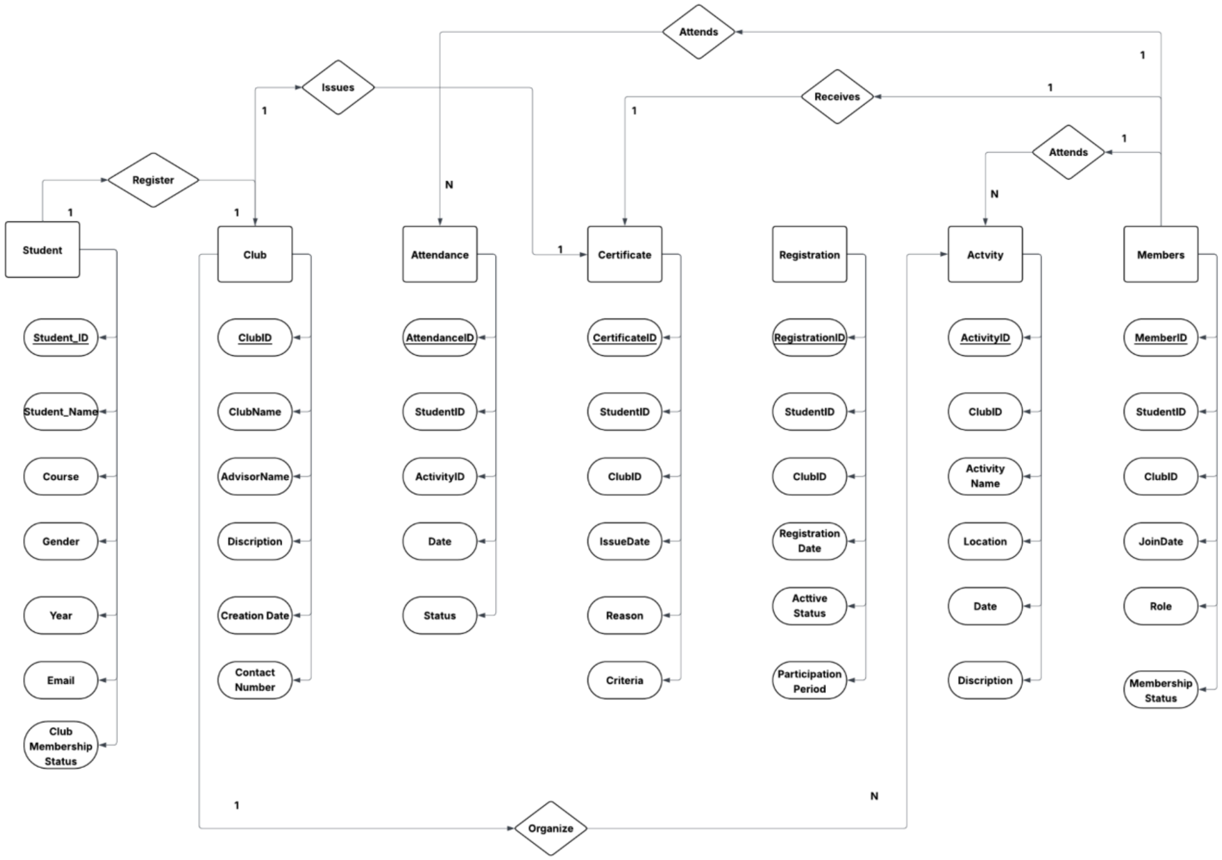
From an operational standpoint, the system will be easy to use for both students and club advisors. Its interface will be designed with simplicity in mind, ensuring that users can perform tasks like club registration, attendance submission, and certificate requests with minimal effort. Training requirements are minimal, as the functionalities are straightforward and intuitive. Furthermore, maintenance of the system will be manageable, with updates and technical support carried out as needed.

Legally, the system will ensure data privacy and secure access by implementing proper authentication mechanisms. Only authorized users will be able to access sensitive information, such as attendance records and certificates. The design will comply with institutional policies and any applicable data protection regulations.

In terms of scheduling, the project is feasible to complete within a reasonable timeframe. With a dedicated team and a clear development plan, the system can be built, tested, and deployed within two to three months. Overall, the Club Registration System is a feasible and beneficial project that meets both institutional and user needs.

6. Database Design

I. Entity-relationship diagram(ER diagram)



Club Management System, this ER diagram shows how data about students, clubs, activities, memberships, attendance, registrations, and certificates is interconnected to support administrative and operational processes.

Entities such as **Student**, **Club**, **Activity**, **Members**, **Attendance**, **Registration**, and **Certificate** are represented using rectangles. Each entity includes various attributes (shown as ovals) such as:

- For Student: Student_ID, Student_Name, Course, Gender, Year, Email, and Club Membership Status.

- For Club: ClubID, ClubName, AdvisorName, Description, Creation Date, and Contact Number.

- For Activity: ActivityID, Activity Name, Location, Date, and Description.

- For Certificate: CertificateID, StudentID, ClubID, IssueDate, Reason, and Criteria.

Key attributes like Student_ID,Club_ID and Activity_ID are underlined to indicate they are unique identifiers for each entity.

Relationships:

Relationships are depicted using diamonds and describe how entities interact. These include:

- **Register**: A student can register for one club.
- **Issues**: A club can issue multiple attendance records.
- **Attends**: A student can attend many activities.
- **Receives**: A student receives certificates from clubs.
- **Organize**: Clubs organize various activities.

Operations Captured:

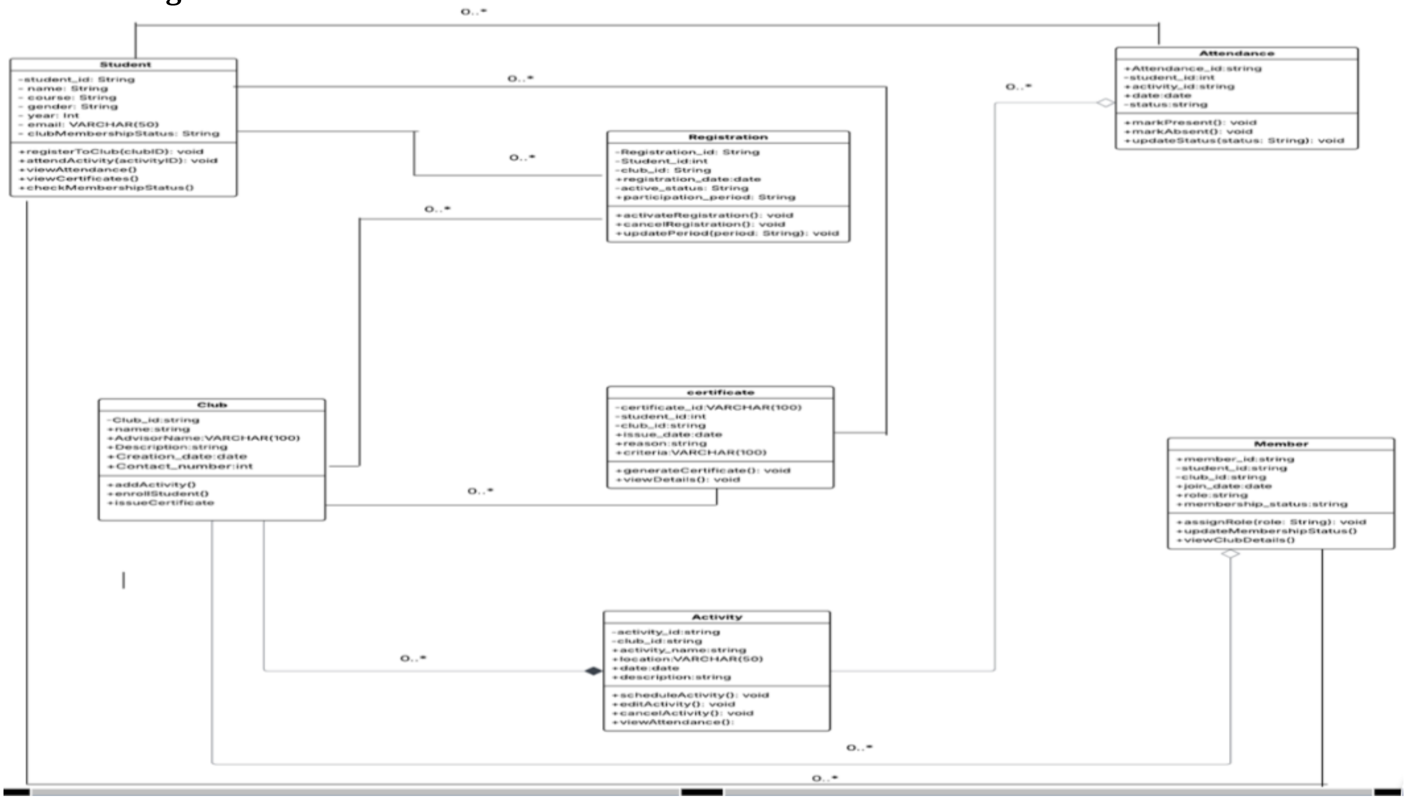
The diagram reflects key operations within the club management environment:

- Students registering and becoming members of various clubs.
- Clubs organizing events and recording student attendance.
- Issuance of certificates to students based on participation or achievements.
- Maintenance of historical data on membership roles and activity involvement.

Cardinalities:

- A student can register for only one club (1:1)
- A club can have many members (1:N)
- An activity can be attended by multiple students (N:M)
- A student can receive only one certificate (1:N)

II.Class Diagram

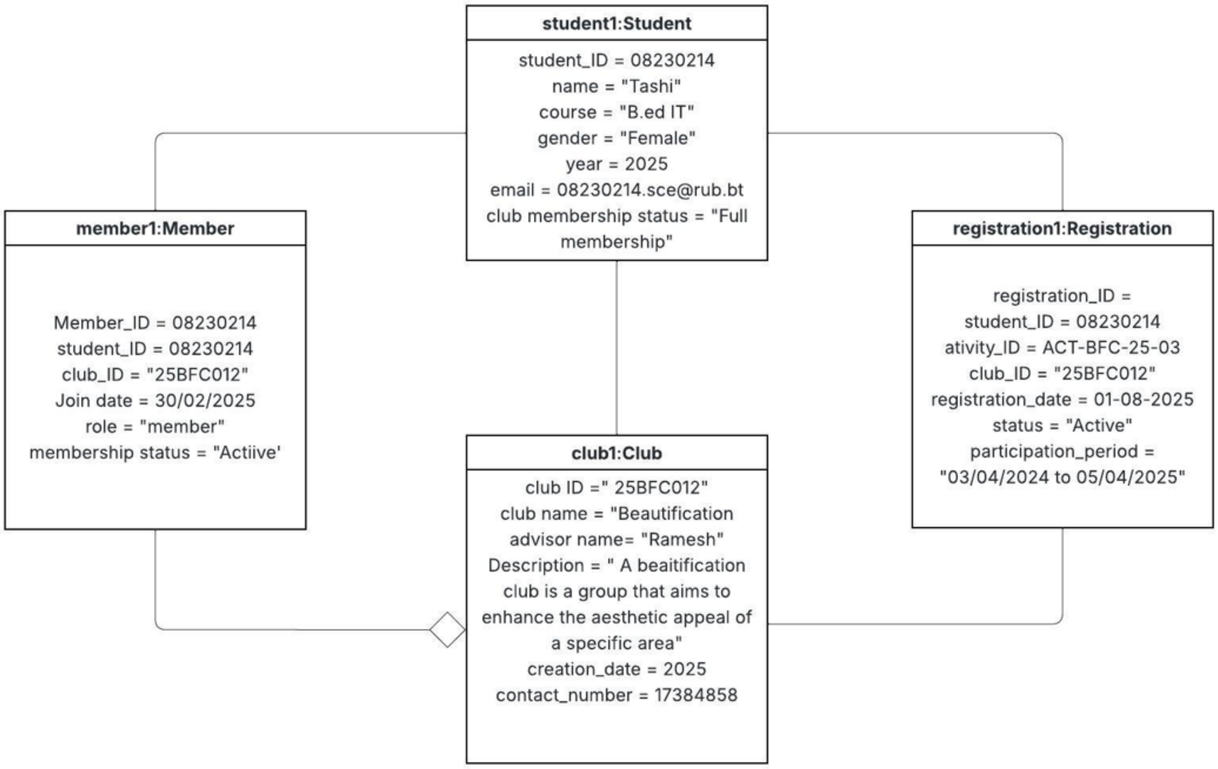


The Class Diagram for the Student Club Management System is a UML static structure diagram that outlines the key classes, their attributes, methods, and relationships in the system. It includes main entities such as Student, Club, Registration, Activity, Attendance, Certificate, and Member. Each class represents a core component of the system. For example, the Student class includes attributes like student_id, name, gender, and email, while the Club class has club_id, name, advisorName, and creation_date.

The diagram shows how students can register for clubs, participate in activities, and receive certificates. The Activity class represents events organized by clubs, and the Attendance class records whether students attended these activities. The Certificate class documents awards or recognitions, and the Member class provides details about a student’s role and membership status in a club.

Relationships are represented with cardinalities, indicating how many objects are associated. For instance, one Club can have many Activities, and one Student can register for multiple Clubs. This class diagram serves as a detailed design model for developing an organized and efficient student club management system.

III. Object Diagram


student1:Student
student_ID = 08230214
name = "Tashi"
course = "B.ed IT"
gender = "Female"
year = 2025
email = 08230214.sce@rub.bt
club membership status = "Full membership"
member1:Member
Member_ID = 08230214
student_ID = 08230214
club_ID = "25BFC012"
Join date = 30/02/2025
role = "member"
membership status = "Actiive'
registration1:Registration
registration_ID =
student_ID = 08230214
ativity_ID = ACT-BFC-25-03
club_ID = "25BFC012"
registration_date = 01-08-2025
status = "Active"
participation_period = "03/04/2024 to 05/04/2025"
club1:Club
club ID =" 25BFC012"
club name = "Beautification
advisor name= "Ramesh"
Description = " A beaitification club is a group that aims to enhance the aesthetic appeal of a specific area"
creation_date = 2025
contact_number = 17384858

This is an **object diagram** representing instances (objects) of different classes in a **Student Club Management System**. Object diagrams show a snapshot of the system at a particular moment, reflecting real data (not just the schema).

student1:Student

This object represents a student named **Tashi**.

- student_ID: 08230214
- name: Tashi
- course: B.Ed IT
- gender: Female

- year: 2025
- email: 08230214.sce@rub.bt
- club membership status: Full membership

club1:Club

This object represents a **Beautification** club.

- **club ID:** 25BFC012
- **club name:** Beautification
- **advisor name:** Ramesh
- **description:** A beautification club that aims to enhance the aesthetic appeal of a specific area
- **creation_date:** 2025
- **contact_number:** 17384858

member1:Member

This object represents **Tashi's membership** in the Beautification club.

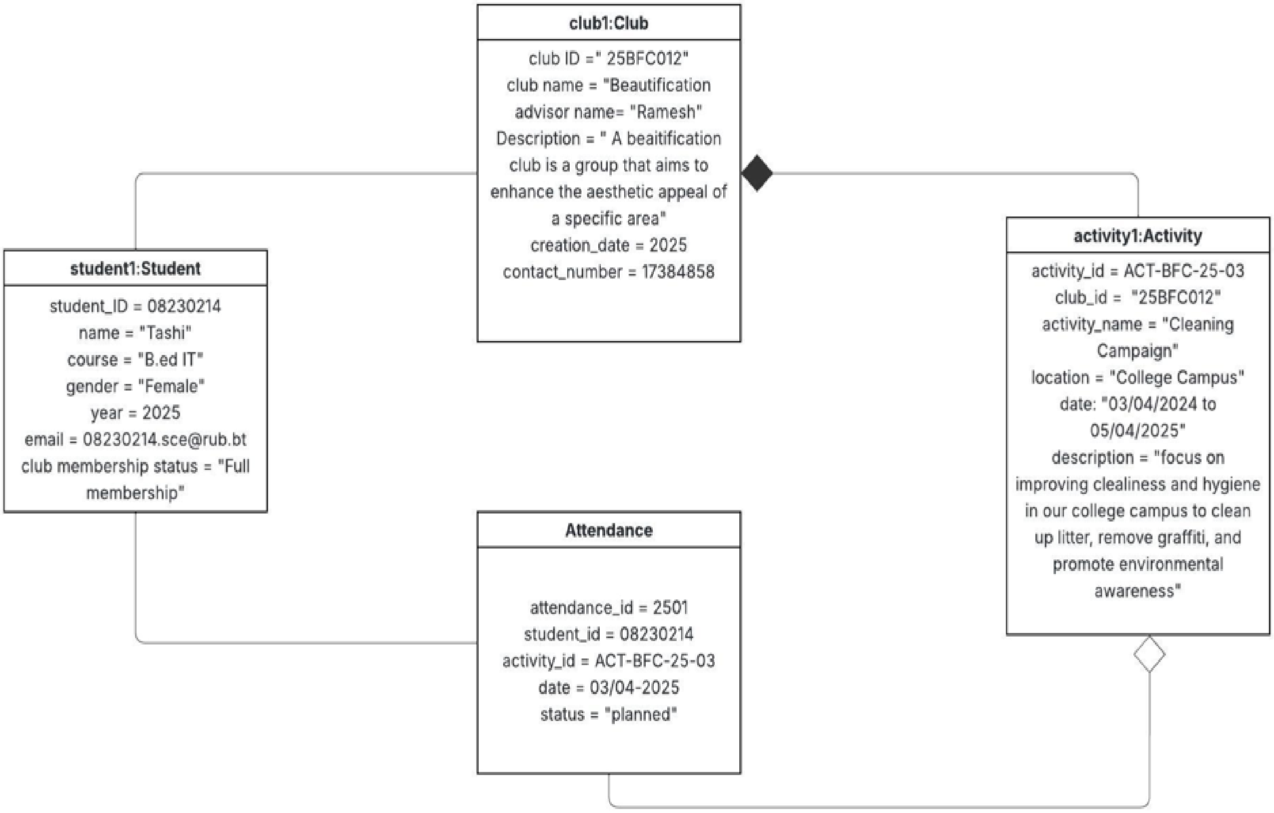
- **Member_ID:** 08230214
- **student_ID:** 08230214
- **club_ID:** 25BFC012
- **Join date:** 30/02/2025
- **role:** member
- **membership status:** 'Active'

registration1:Registration

This object represents **Tashi's registration** for a club activity.

- **registration_ID:** 08230214

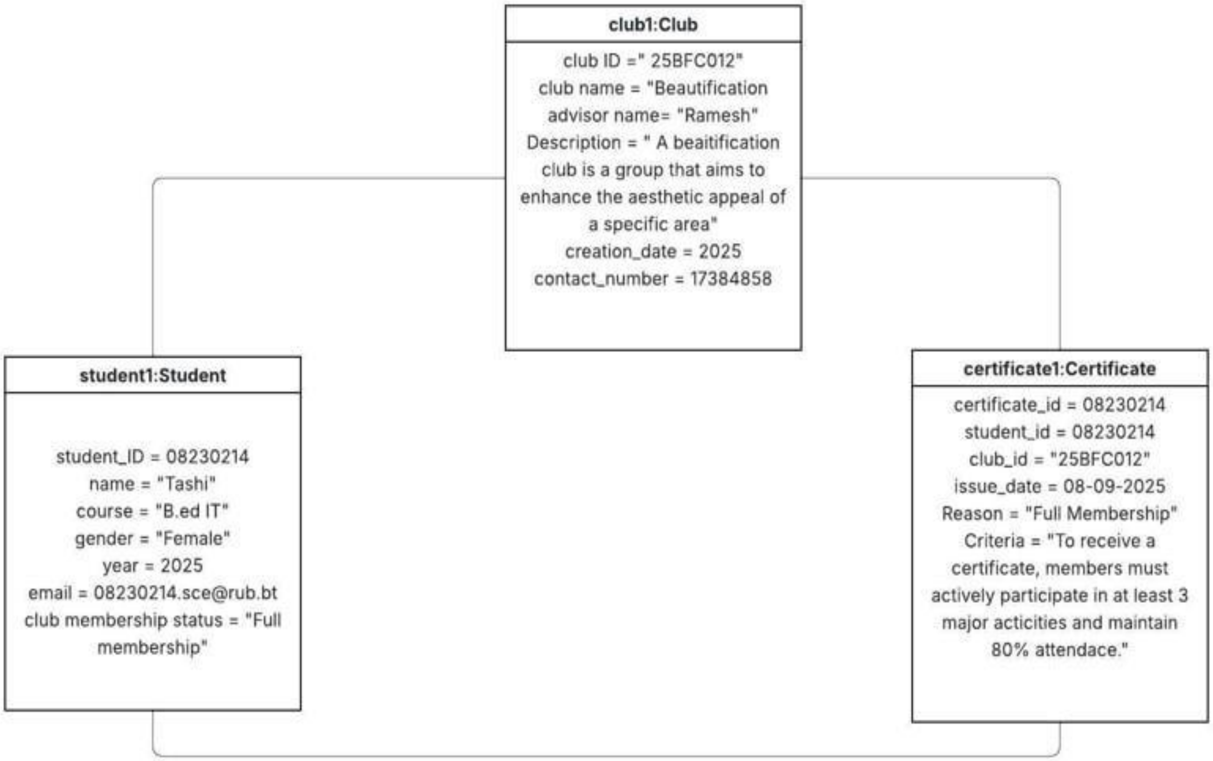
- **student_ID:** 08230214
- **activity_ID:** ACT-BFC-25-03
- **club_ID:** 25BFC012
- **registration_date:** 01-08-2025
- **status:** Active
- **participation_period:** 03/04/2024 to 05/04/2025
- One **Student** (Tashi) is a **Member** of one **Club** (Beautification).
- The **Student** has **Registered** for an **Activity** organized by that Club.
- All objects are interconnected through consistent use of student_ID and club_ID.


club1:Club
club ID =" 25BFC012"
club name = "Beautification
advisor name= "Ramesh"
Description = " A beaitification club is a group that aims to enhance the aesthetic appeal of a specific area"
creation_date = 2025
contact_number = 17384858
student1:Student
student_ID = 08230214
name = "Tashi"
course = "B.ed IT"
gender = "Female"
year = 2025
email = 08230214.sce@rub.bt
club membership status = "Full membership"
activity1:Activity
activity_id = ACT-BFC-25-03
club_id = "25BFC012"
activity_name = "Cleaning Campaign"
location = "College Campus"
date: "03/04/2024 to 05/04/2025"
description = "focus on improving clealiness and hygiene in our college campus to clean up litter, remove graffiti, and promote environmental awareness"
Attendance
attendance_id = 2501
student_id = 08230214
activity_id = ACT-BFC-25-03
date = 03/04-2025
status = "planned"

The above object diagram shows the relationship between a student, a club, an activity, and attendance:

- **Tashi**, a B.Ed IT student, is a **full member** of the **Beautification Club**.
- The **Beautification Club**, created in 2025 and advised by **Ramesh**, focuses on improving the aesthetic appeal of areas.
- The club organizes activities, such as the **"Cleaning Campaign"**, which takes place on the **college campus** from **03/04/2024 to 05/04/2025**.
- Tashi is linked to this activity through an **Attendance** record, which shows she is **planned** to participate on **03/04/2025**.

The above object diagram highlights how a student joins a club, participates in its activities, and how attendance is tracked.


club1:Club
club ID =" 25BFC012"
club name = "Beautification
advisor name= "Ramesh"
Description = " A beaitification club is a group that aims to enhance the aesthetic appeal of a specific area"
creation_date = 2025
contact_number = 17384858
student1:Student
student_ID = 08230214
name = "Tashi"
course = "B.ed IT"
gender = "Female"
year = 2025
email = 08230214.sce@rub.bt
club membership status = "Full membership"
certificate1:Certificate
certificate_id = 08230214
student_id = 08230214
club_id = "25BFC012"
issue_date = 08-09-2025
Reason = "Full Membership"
Criteria = "To receive a certificate, members must actively participate in at least 3 major acticities and maintain 80% attendace."

The above object diagram shows the relationship between a student, a club, an activity, and attendance:

- **Tashi**, a B.Ed IT student, is a **full member** of the **Beautification Club**.

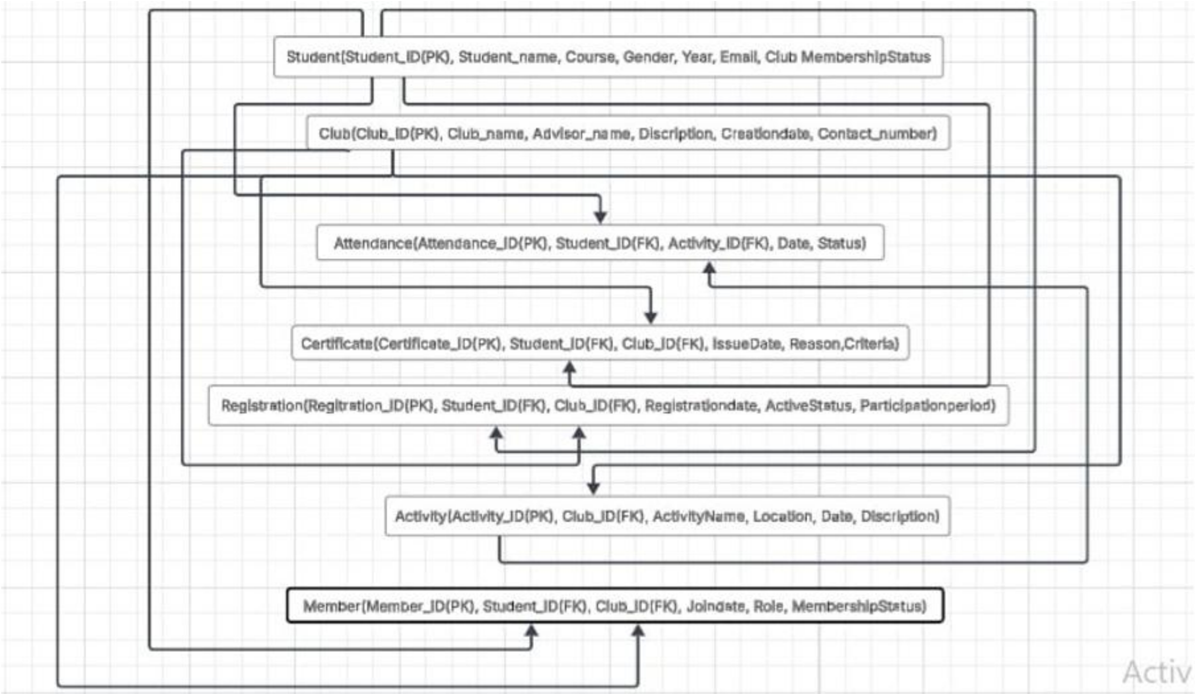
- The **Beautification Club**, created in 2025 and advised by **Ramesh**, focuses on improving the aesthetic appeal of areas.

- The club organizes activities, such as the **"Cleaning Campaign"**, which takes place on the **college campus** from **03/04/2024 to 05/04/2025**.

- Tashi is linked to this activity through an **Attendance** record, which shows she is **planned** to participate on **03/04/2025**.

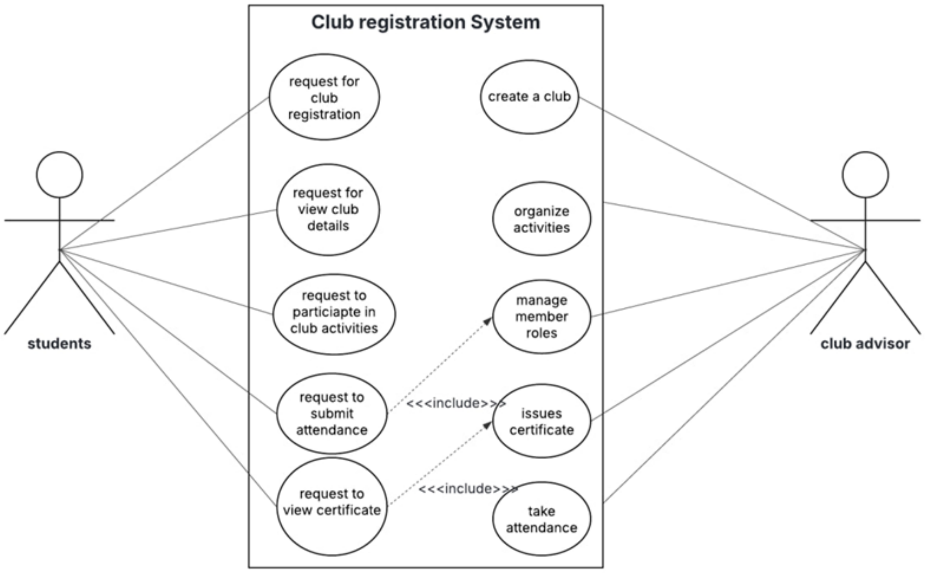
This object diagram highlights how a student joins a club, participates in its activities, and how attendance is tracked.

IV. Relational Diagram


Student(Student_ID(PK), Student_name, Course, Gender, Year, Email, Club MembershipStatus
Club(Club_ID(PK), Club_name, Advisor_name, Discription, Creationdate, Contact_number)
Attendance(Attendance_ID(PK), Student_ID(FK), Activity_ID(FK), Date, Status)
Certificate(Certificate_ID(PK), Student_ID(FK), Club_ID(FK), IssueDate, Reason,Criteria)
Registration(Regitration_ID(PK), Student_ID(FK), Club_ID(FK), Registrationdate, ActiveStatus, Participationperiod)
Activity(Activity_ID(PK), Club_ID(FK), ActivityName, Location, Date, Discription)
Member(Member_ID(PK), Student_ID(FK), Club_ID(FK), Joindate, Role, MembershipStatus)

This relational diagram represents a Student Club Management System consisting of entities like Student, Club, Activity, Attendance, Certificate, Registration, and Member. Each student can register for and participate in multiple clubs and activities, with their attendance and certificates recorded accordingly. Clubs organize activities and maintain information about student members, including their roles and membership status. Relationships between entities are managed using primary and foreign keys, enabling efficient tracking of student involvement, club operations, and related data within the system.

V. Use-case diagram


Club registration System
request for club registration
request for view club details
request to particiapte in club activities
request to submit attendance
request to view certificate
create a club
organize activities
manage member roles
issues certificate
take attendance
<<<include>>>
<<<include>>>
students
club advisor

The Club Registration System is designed to streamline interactions between students and club advisors through a structured set of functionalities. Students are empowered to engage with the system by requesting club registrations, viewing detailed club information, submitting their attendance, participating in club activities, and accessing their certificates. On the other side, club advisors take on an administrative role, creating clubs, organizing events, managing member roles, taking attendance, and issuing certificates. The system ensures a smooth flow of activities, where certain actions are interdependent—for example, viewing a certificate requires it to be issued, and submitting attendance relies on the advisor's verification. This use case model provides a clear and organized representation of how users interact with the system, ensuring that both participation and management are seamlessly integrated.

Normalization

Our Club Management System consists of seven main entities:

- Student
- Club
- Members
- Registration
- Certificate
- Attendance
- Activity

Among these, the Student, Club, Members, Registration, and Certificate tables are already in normalized form. However, the Attendance and Activity table initially contained multi-valued attributes and repeating groups, so we applied normalization steps (1NF → 2NF → 3NF) specifically to it.

Attendance Table

In 1NF, we made sure that all values in the table are atomic (indivisible), and there are no repeating groups.

Before 1NF (Not Valid)

- The Status column in the attendance table could potentially store multiple attendance states like “Present but Late” or “Absent but Excused” in a single cell, which violates the 1NF rule of atomicity.

- Each cell must contain a single, indivisible value, but storing multiple statues in one field makes it hard to filter, count, or analyze attendance data accurately.

Attendance_ID	Student_ID	Activity_ID	Date	Status
414	08230232	314	2025-04-27	Present
415	08230236	315	2025-04-28	Present Late
416	08230253	316	2025-04-27	Absent Excused
417	08230231	317	2025-04-27	Present
418	08230222	318	2025-04-27	Present

After 1NF (valid)

We decomposed the multivalued attributes so each row now represents one specific status.

All columns contain only atomic values.

Repeating groups have been eliminated.

Attendance_ID	Student_ID	Activity_ID	Date	Status
414	08230232	314	2025-04-27	Present
415	08230236	315	2025-04-28	Present
415	08230236	315	2025-04-28	Late
416	08230253	316	2025-04-27	Absent
416	08230253	316	2025-04-27	Excused
417	08230231	317	2025-04-27	Present
418	08230222	318	2025-04-27	Present

In the **1NF version**, if Attendance_ID is not a surrogate key and the primary key is a composite key (Student_ID, Activity_ID), then columns like Date and Status depend only onActivity_ID not the entire key.

So, we decompose the table into two separate tables to ensure all non-key attributes are fully functionally dependent on the entire key.

1. Attendance Table

This table keeps the relationship between students and activities:

Attendance_ID	Student_ID	Activity_ID
414	08230232	314
415	08230236	315
416	08230253	316
417	08230231	317
418	08230222	318

2. ActivityAttendanceDetails Table

This stores activity-specific attendance details like Date and status

Attendance_ID	Date	Status
414	2025-04-27	Present
415	2025-04-28	Present
415	2025-04-28	Late
416	2025-04-27	Absent
416	2025-04-27	Excused
417	2025-04-27	Present
418	2025-04-27	Present

- ➢ All non-key attributes in both tables now **depend on the whole primary key** (in each table).
- ➢ Status and Date are now tied to a specific Attendance_ID (a single unique row), not a partial key like Activity_ID.

The Attendance table contains simple and atomic fields, and no significant transitive dependencies exist. Attributes like Status and Date are straightforward and do not require further decomposition. Stopping at 2NF keeps the database structure efficient, easier to query, and maintainable, while still reducing redundancy and ensuring data integrity. Given the scope and simplicity of the system, further normalization to 3NF would add unnecessary complexity without substantial benefit.

Activity Table

The Activity table might contain repeating groups or multivalued attributes in one column.
 A single row store multiple activities or locations like:

ActivityID	ClubID	ActivityName	Location	Date	Description
301	1	Cleaning, Decorating	Guest House	2025-05-28	Done
302	2	Praying	Prayer Hall	2025-05-29	Done
303	3	Competition	Football Ground	2025-06-5	Done

- ❖ **This violates the rule of atomicity — each cell must hold a single value only.**

After 1NF (Valid)

We decompose multivalued attributes so that:

- Each activity appears on a separate row.
- All columns contain only atomic values.
- No repeating groups exist.

ActivityID	ClubID	ActivityName	Location	Date	Description
301	1	Cleaning	Guest House	2025-05-28	Done
301	1	Decorating	Guest House	2025-05-28	Done
302	2	Praying	Prayer Hall	2025-05-29	Done
303	3	Competition	Football Ground	2025-06-5	Done

After 2NF (Partial Dependency Removed)

If the primary key were composite (ActivityID and ClubID) and attributes like ActivityName , Location, Date and Description depend only on ActivityID, then partial dependency exists.

To fix this, we:

- Separate the table into:
- A **ClubActivity** table (just linking clubs and activities)
- An **ActivityDetails** table (storing activity-specific info)

❖ **ClubActivity**

ActivityID	ClubID
301	1
302	2
303	3

❖ **ActivityDetails**

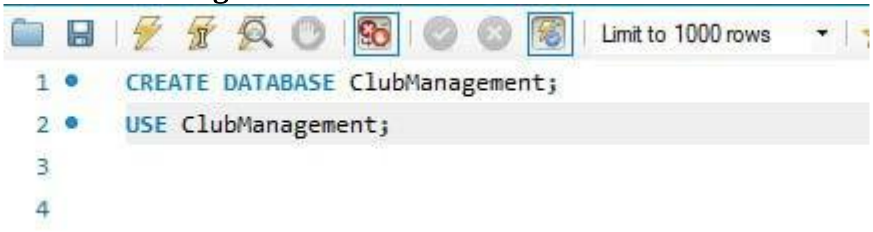
ActivityID	ActivityName	Location	Date	Description
301	Cleaning	Guest House	2025-05-28	Done
301	Decorating	Guest House	2025-05-28	Done
302	Praying	Prayer Hall	2025-05-29	Done
303	Competition	Football Ground	2025-06-5	Done

In this case normalization up to 2NF is sufficient because:

- Locations are simple strings with no additional attributes.
- The activity structure is already clear and atomic.
- Further decomposition into 3NF (e.g., creating a Location table) adds unnecessary complexity without meaningful gain.

The Activity table in the Club Management System contains straightforward attributes like ActivityName, Location, Date and Description. These are atomic and directly dependent on the primary key. Normalizing up to Second Normal Form (2NF) eliminates partial dependencies and ensures data integrity. Given the simplicity of the Location field and the low risk of redundancy, further normalization to Third Normal Form (3NF) is not necessary at this stage and would introduce additional complexity without significant benefit.

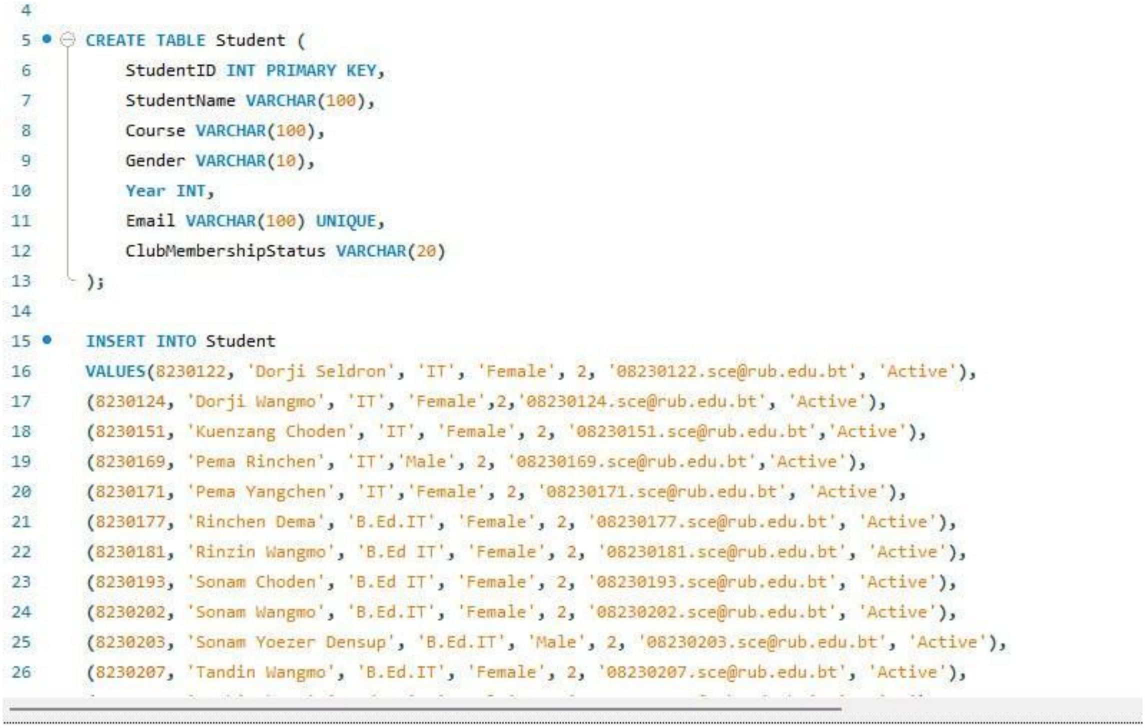
1. Creating Databases



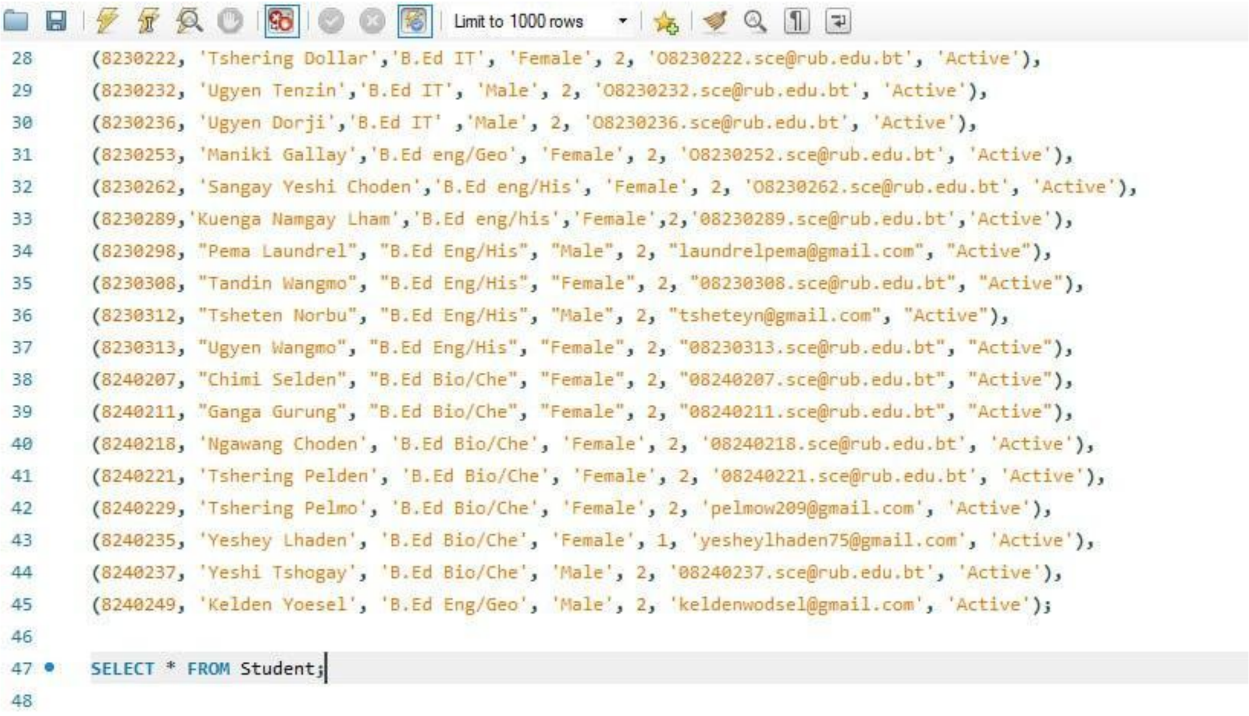
```
CREATE DATABASE ClubManagement;
USE ClubManagement;
```

To begin developing the Club Management System, the first step was to create a dedicated database using the SQL command CREATE DATABASE ClubManagement;. This command initializes a new database named *ClubManagement*, which will serve as the central repository for all related data, including member information, club events, and financial records. Following the creation of the database, the USE ClubManagement; command was executed to select and activate this database as the working environment. This ensures that all subsequent operations, such as creating tables and inserting data, are applied specifically within the *ClubManagement* database. This foundational step is essential for organizing data efficiently and maintaining a structured environment for further development of the system.

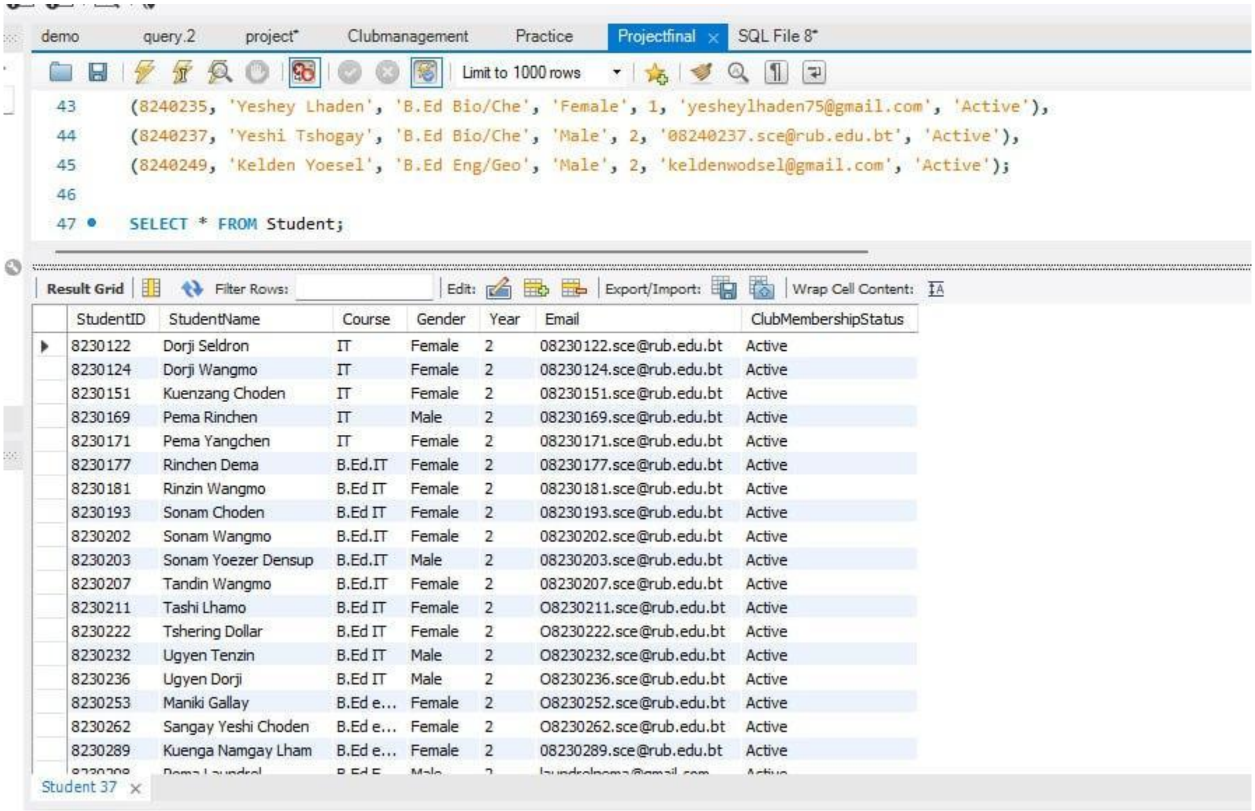
2. Student table



```sql

CREATE TABLE Student (
    StudentID INT PRIMARY KEY,
    StudentName VARCHAR(100),
    Course VARCHAR(100),
    Gender VARCHAR(10),
    Year INT,
    Email VARCHAR(100) UNIQUE,
    ClubMembershipStatus VARCHAR(20)
);

INSERT INTO Student
VALUES(8230122, 'Dorji Seldron', 'IT', 'Female', 2, '08230122.sce@rub.edu.bt', 'Active'),
(8230124, 'Dorji Wangmo', 'IT', 'Female',2,'08230124.sce@rub.edu.bt', 'Active'),
(8230151, 'Kuenzang Choden', 'IT', 'Female', 2, '08230151.sce@rub.edu.bt','Active'),
(8230169, 'Pema Rinchen', 'IT','Male', 2, '08230169.sce@rub.edu.bt','Active'),
(8230171, 'Pema Yangchen', 'IT','Female', 2, '08230171.sce@rub.edu.bt', 'Active'),
(8230177, 'Rinchen Dema', 'B.Ed.IT', 'Female', 2, '08230177.sce@rub.edu.bt', 'Active'),
(8230181, 'Rinzin Wangmo', 'B.Ed IT', 'Female', 2, '08230181.sce@rub.edu.bt', 'Active'),
(8230193, 'Sonam Choden', 'B.Ed IT', 'Female', 2, '08230193.sce@rub.edu.bt', 'Active'),
(8230202, 'Sonam Wangmo', 'B.Ed.IT', 'Female', 2, '08230202.sce@rub.edu.bt', 'Active'),
(8230203, 'Sonam Yoezer Densup', 'B.Ed.IT', 'Male', 2, '08230203.sce@rub.edu.bt', 'Active'),
(8230207, 'Tandin Wangmo', 'B.Ed.IT', 'Female', 2, '08230207.sce@rub.edu.bt', 'Active'),
```



```sql
(8230222, 'Tshering Dollar','B.Ed IT', 'Female', 2, 'O8230222.sce@rub.edu.bt', 'Active'),
(8230232, 'Ugyen Tenzin','B.Ed IT', 'Male', 2, 'O8230232.sce@rub.edu.bt', 'Active'),
(8230236, 'Ugyen Dorji','B.Ed IT' ,'Male', 2, 'O8230236.sce@rub.edu.bt', 'Active'),
(8230253, 'Maniki Gallay','B.Ed eng/Geo', 'Female', 2, 'O8230252.sce@rub.edu.bt', 'Active'),
(8230262, 'Sangay Yeshi Choden','B.Ed eng/His', 'Female', 2, 'O8230262.sce@rub.edu.bt', 'Active'),
(8230289,'Kuenga Namgay Lham','B.Ed eng/his','Female',2,'08230289.sce@rub.edu.bt','Active'),
(8230298, "Pema Laundrel", "B.Ed Eng/His", "Male", 2, "laundrelpema@gmail.com", "Active"),
(8230308, "Tandin Wangmo", "B.Ed Eng/His", "Female", 2, "08230308.sce@rub.edu.bt", "Active"),
(8230312, "Tsheten Norbu", "B.Ed Eng/His", "Male", 2, "tsheteyn@gmail.com", "Active"),
(8230313, "Ugyen Wangmo", "B.Ed Eng/His", "Female", 2, "08230313.sce@rub.edu.bt", "Active"),
(8240207, "Chimi Selden", "B.Ed Bio/Che", "Female", 2, "08240207.sce@rub.edu.bt", "Active"),
(8240211, "Ganga Gurung", "B.Ed Bio/Che", "Female", 2, "08240211.sce@rub.edu.bt", "Active"),
(8240218, 'Ngawang Choden', 'B.Ed Bio/Che', 'Female', 2, '08240218.sce@rub.edu.bt', 'Active'),
(8240221, 'Tshering Pelden', 'B.Ed Bio/Che', 'Female', 2, '08240221.sce@rub.edu.bt', 'Active'),
(8240229, 'Tshering Pelmo', 'B.Ed Bio/Che', 'Female', 2, 'pelmow209@gmail.com', 'Active'),
(8240235, 'Yeshey Lhaden', 'B.Ed Bio/Che', 'Female', 1, 'yesheylhaden75@gmail.com', 'Active'),
(8240237, 'Yeshi Tshogay', 'B.Ed Bio/Che', 'Male', 2, '08240237.sce@rub.edu.bt', 'Active'),
(8240249, 'Kelden Yoesel', 'B.Ed Eng/Geo', 'Male', 2, 'keldenwodsel@gmail.com', 'Active');

SELECT * FROM Student;

```

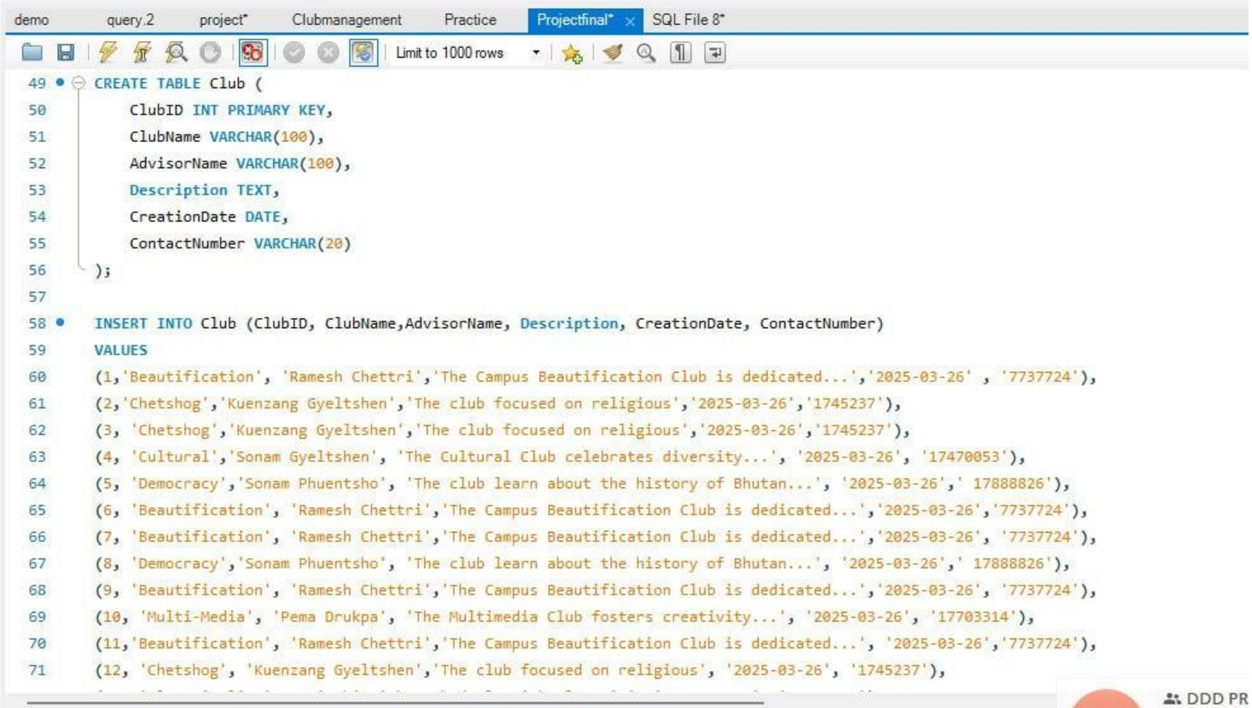


```
(8240235, 'Yeshey Lhaden', 'B.Ed Bio/Che', 'Female', 1, 'yesheylhaden75@gmail.com', 'Active'),
(8240237, 'Yeshi Tshogay', 'B.Ed Bio/Che', 'Male', 2, '08240237.sce@rub.edu.bt', 'Active'),
(8240249, 'Kelden Yoesel', 'B.Ed Eng/Geo', 'Male', 2, 'keldenwodsel@gmail.com', 'Active');

SELECT * FROM Student;
```

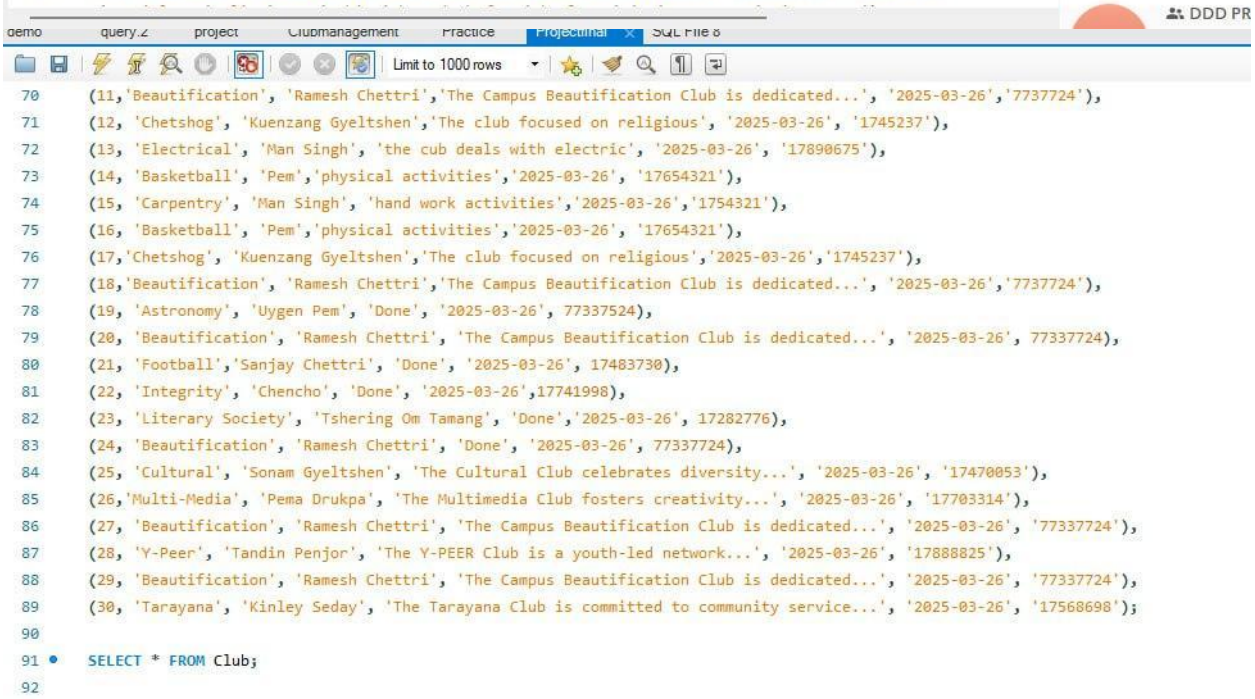
StudentID	StudentName	Course	Gender	Year	Email	ClubMembershipStatus
8230122	Dorji Seldron	IT	Female	2	08230122.sce@rub.edu.bt	Active
8230124	Dorji Wangmo	IT	Female	2	08230124.sce@rub.edu.bt	Active
8230151	Kuenzang Choden	IT	Female	2	08230151.sce@rub.edu.bt	Active
8230169	Pema Rinchen	IT	Male	2	08230169.sce@rub.edu.bt	Active
8230171	Pema Yangchen	IT	Female	2	08230171.sce@rub.edu.bt	Active
8230177	Rinchen Dema	B.Ed.IT	Female	2	08230177.sce@rub.edu.bt	Active
8230181	Rinzin Wangmo	B.Ed IT	Female	2	08230181.sce@rub.edu.bt	Active
8230193	Sonam Choden	B.Ed IT	Female	2	08230193.sce@rub.edu.bt	Active
8230202	Sonam Wangmo	B.Ed.IT	Female	2	08230202.sce@rub.edu.bt	Active
8230203	Sonam Yoezer Densup	B.Ed.IT	Male	2	08230203.sce@rub.edu.bt	Active
8230207	Tandin Wangmo	B.Ed.IT	Female	2	08230207.sce@rub.edu.bt	Active
8230211	Tashi Lhamo	B.Ed IT	Female	2	O8230211.sce@rub.edu.bt	Active
8230222	Tshering Dollar	B.Ed IT	Female	2	O8230222.sce@rub.edu.bt	Active
8230232	Ugyen Tenzin	B.Ed IT	Male	2	O8230232.sce@rub.edu.bt	Active
8230236	Ugyen Dorji	B.Ed IT	Male	2	O8230236.sce@rub.edu.bt	Active
8230253	Maniki Gallay	B.Ed e...	Female	2	O8230252.sce@rub.edu.bt	Active
8230262	Sangay Yeshi Choden	B.Ed e...	Female	2	O8230262.sce@rub.edu.bt	Active
8230289	Kuenga Namgay Lham	B.Ed e...	Female	2	08230289.sce@rub.edu.bt	Active

The Student table was created to organize and manage detailed information about students participating in club activities. It includes important fields such as StudentID to uniquely identify each student, StudentName for their full names, Course to indicate their academic program, Gender, Year of study, and Email which is constrained to be unique to prevent duplicate entries. Additionally, the table tracks the ClubMembershipStatus to show whether a student is actively involved in the club. Following the table's creation, 30 student records were inserted, each containing complete and consistent information across all fields. When the SELECT * FROM Student; command is executed, it retrieves all the records, displaying a comprehensive list of students with their corresponding details. This output confirms that the data was successfully stored and can now be accessed or managed easily for club-related activities such as communication, membership verification, and reporting.

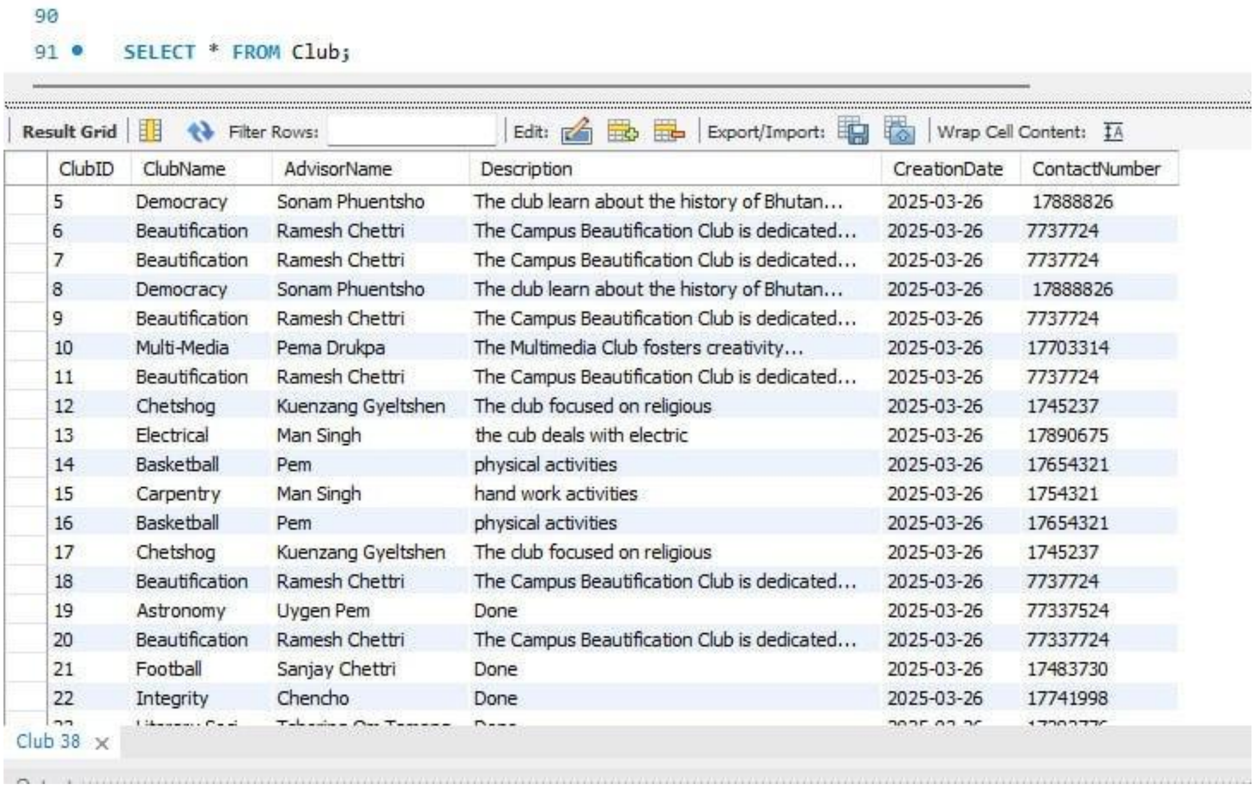
3. Club table



```sql
CREATE TABLE Club (
    ClubID INT PRIMARY KEY,
    ClubName VARCHAR(100),
    AdvisorName VARCHAR(100),
    Description TEXT,
    CreationDate DATE,
    ContactNumber VARCHAR(20)
);

INSERT INTO Club (ClubID, ClubName,AdvisorName, Description, CreationDate, ContactNumber)
VALUES
(1,'Beautification', 'Ramesh Chettri','The Campus Beautification Club is dedicated...','2025-03-26' , '7737724'),
(2,'Chetshog','Kuenzang Gyeltshen','The club focused on religious','2025-03-26','1745237'),
(3, 'Chetshog','Kuenzang Gyeltshen','The club focused on religious','2025-03-26','1745237'),
(4, 'Cultural','Sonam Gyeltshen', 'The Cultural Club celebrates diversity...', '2025-03-26', '17470053'),
(5, 'Democracy','Sonam Phuentsho', 'The club learn about the history of Bhutan...', '2025-03-26',' 17888826'),
(6, 'Beautification', 'Ramesh Chettri','The Campus Beautification Club is dedicated...','2025-03-26','7737724'),
(7, 'Beautification', 'Ramesh Chettri','The Campus Beautification Club is dedicated...','2025-03-26', '7737724'),
(8, 'Democracy','Sonam Phuentsho', 'The club learn about the history of Bhutan...', '2025-03-26',' 17888826'),
(9, 'Beautification', 'Ramesh Chettri','The Campus Beautification Club is dedicated...','2025-03-26', '7737724'),
(10, 'Multi-Media', 'Pema Drukpa', 'The Multimedia Club fosters creativity...', '2025-03-26', '17703314'),
(11,'Beautification', 'Ramesh Chettri','The Campus Beautification Club is dedicated...', '2025-03-26','7737724'),
(12, 'Chetshog', 'Kuenzang Gyeltshen','The club focused on religious', '2025-03-26', '1745237'),
```



```sql
(11,'Beautification', 'Ramesh Chettri','The Campus Beautification Club is dedicated...', '2025-03-26','7737724'),
(12, 'Chetshog', 'Kuenzang Gyeltshen','The club focused on religious', '2025-03-26', '1745237'),
(13, 'Electrical', 'Man Singh', 'the cub deals with electric', '2025-03-26', '17890675'),
(14, 'Basketball', 'Pem','physical activities','2025-03-26', '17654321'),
(15, 'Carpentry', 'Man Singh', 'hand work activities','2025-03-26','1754321'),
(16, 'Basketball', 'Pem','physical activities','2025-03-26', '17654321'),
(17,'Chetshog', 'Kuenzang Gyeltshen','The club focused on religious','2025-03-26','1745237'),
(18,'Beautification', 'Ramesh Chettri','The Campus Beautification Club is dedicated...', '2025-03-26','7737724'),
(19, 'Astronomy', 'Uygen Pem', 'Done', '2025-03-26', 77337524),
(20, 'Beautification', 'Ramesh Chettri', 'The Campus Beautification Club is dedicated...', '2025-03-26', 77337724),
(21, 'Football','Sanjay Chettri', 'Done', '2025-03-26', 17483730),
(22, 'Integrity', 'Chencho', 'Done', '2025-03-26',17741998),
(23, 'Literary Society', 'Tshering Om Tamang', 'Done','2025-03-26', 17282776),
(24, 'Beautification', 'Ramesh Chettri', 'Done', '2025-03-26', 77337724),
(25, 'Cultural', 'Sonam Gyeltshen', 'The Cultural Club celebrates diversity...', '2025-03-26', '17470053'),
(26,'Multi-Media', 'Pema Drukpa', 'The Multimedia Club fosters creativity...', '2025-03-26', '17703314'),
(27, 'Beautification', 'Ramesh Chettri', 'The Campus Beautification Club is dedicated...', '2025-03-26', '77337724'),
(28, 'Y-Peer', 'Tandin Penjor', 'The Y-PEER Club is a youth-led network...', '2025-03-26', '17888825'),
(29, 'Beautification', 'Ramesh Chettri', 'The Campus Beautification Club is dedicated...', '2025-03-26', '77337724'),
(30, 'Tarayana', 'Kinley Seday', 'The Tarayana Club is committed to community service...', '2025-03-26', '17568698');

SELECT * FROM Club;
```

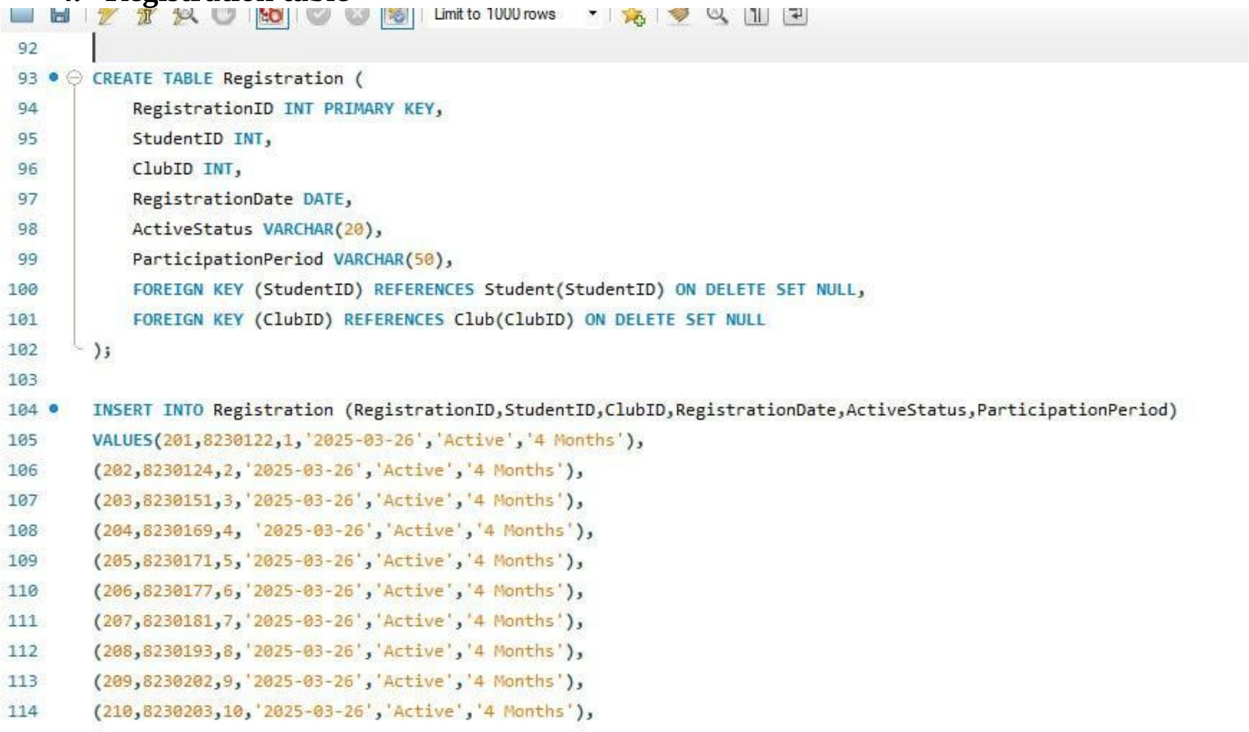


```
SELECT * FROM Club;
```

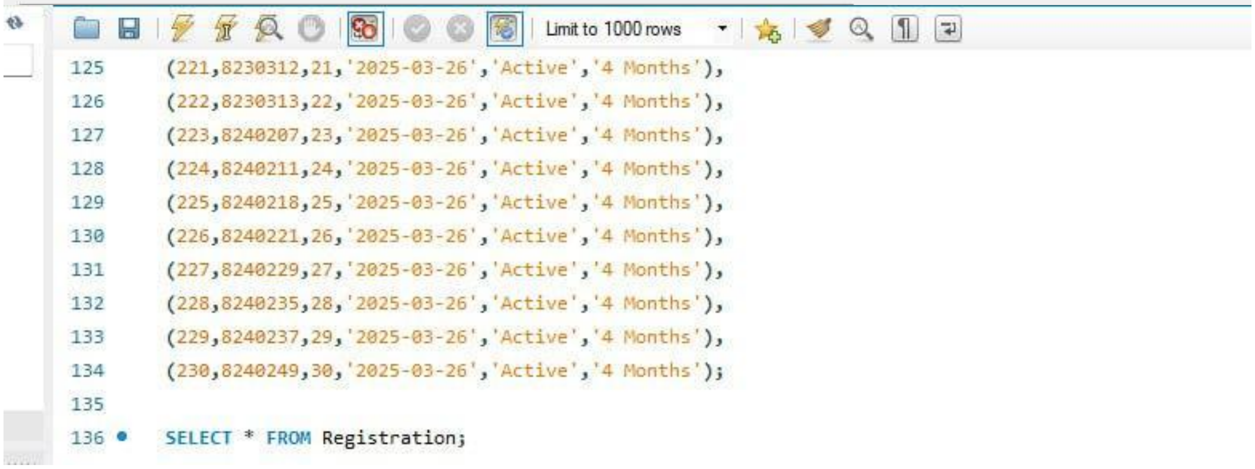
ClubID	ClubName	AdvisorName	Description	CreationDate	ContactNumber
5	Democracy	Sonam Phuentsho	The club learn about the history of Bhutan...	2025-03-26	17888826
6	Beautification	Ramesh Chettri	The Campus Beautification Club is dedicated...	2025-03-26	7737724
7	Beautification	Ramesh Chettri	The Campus Beautification Club is dedicated...	2025-03-26	7737724
8	Democracy	Sonam Phuentsho	The club learn about the history of Bhutan...	2025-03-26	17888826
9	Beautification	Ramesh Chettri	The Campus Beautification Club is dedicated...	2025-03-26	7737724
10	Multi-Media	Pema Drukpa	The Multimedia Club fosters creativity...	2025-03-26	17703314
11	Beautification	Ramesh Chettri	The Campus Beautification Club is dedicated...	2025-03-26	7737724
12	Chetshog	Kuenzang Gyeltshen	The club focused on religious	2025-03-26	1745237
13	Electrical	Man Singh	the cub deals with electric	2025-03-26	17890675
14	Basketball	Pem	physical activities	2025-03-26	17654321
15	Carpentry	Man Singh	hand work activities	2025-03-26	1754321
16	Basketball	Pem	physical activities	2025-03-26	17654321
17	Chetshog	Kuenzang Gyeltshen	The club focused on religious	2025-03-26	1745237
18	Beautification	Ramesh Chettri	The Campus Beautification Club is dedicated...	2025-03-26	7737724
19	Astronomy	Uygen Pem	Done	2025-03-26	77337524
20	Beautification	Ramesh Chettri	The Campus Beautification Club is dedicated...	2025-03-26	77337724
21	Football	Sanjay Chettri	Done	2025-03-26	17483730
22	Integrity	Chencho	Done	2025-03-26	17741998

The **Club** table is designed to store detailed information about various clubs within the organization. It consists of several fields: ClubID, which uniquely identifies each club and serves as the primary key; ClubName representing the name of the club; AdvisorName indicating the name of the faculty advisor responsible for the club; Description providing a textual overview of the club's purpose and activities; CreationDate to record when the club was begin in the semester; and ContactNumber for the club's communication. After the table structure was defined, 30 records were inserted, each corresponding to different clubs and their details. Some clubs, such as "Beautification" and "Chetshog," appear multiple times with identical or slightly different details, which may indicate repeated entries or separate branches/chapters of the same club. The data includes varied club types ranging from cultural and sports to academic and community service clubs, showcasing diversity in student engagement. When the SELECT * FROM Club; command is executed, it retrieves all records, displaying a complete list of clubs with their associated information, which confirms the data is stored correctly and is ready for management and analysis purposes.

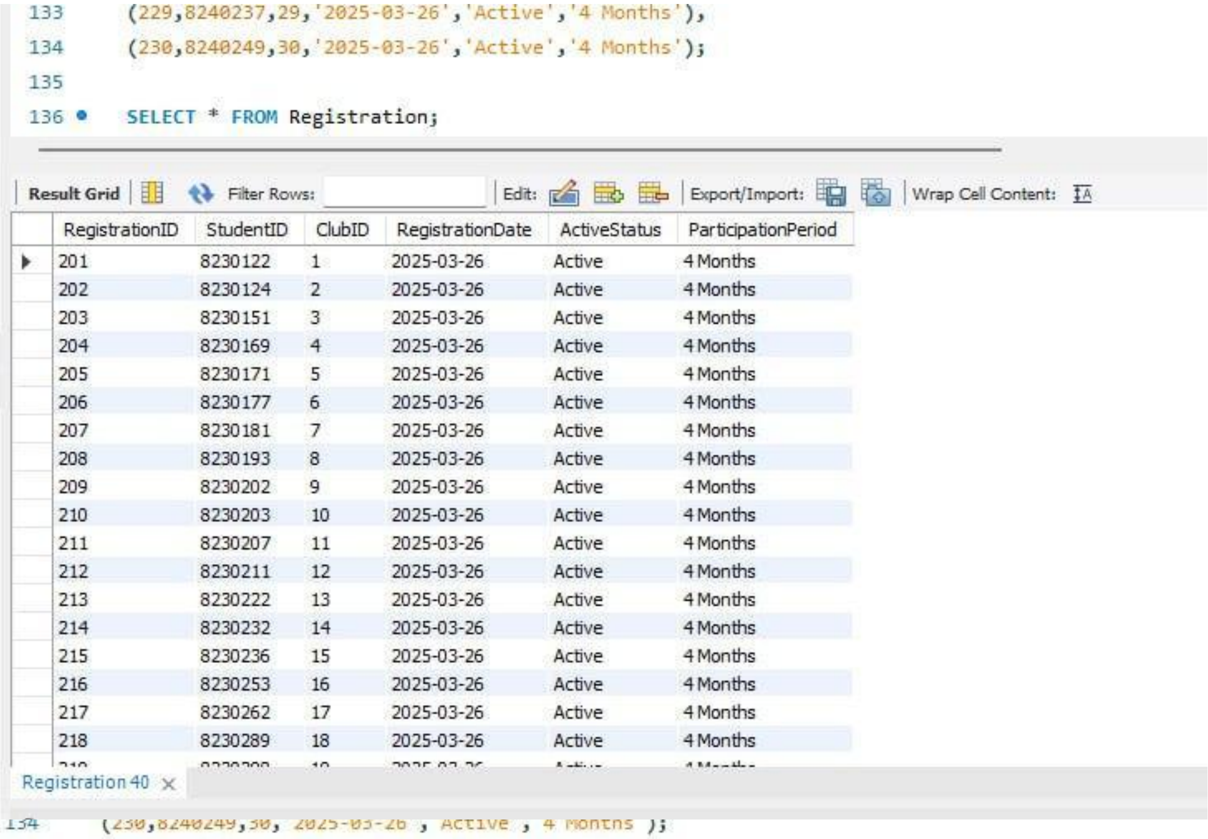
4. Registration table



```
CREATE TABLE Registration (
    RegistrationID INT PRIMARY KEY,
    StudentID INT,
    ClubID INT,
    RegistrationDate DATE,
    ActiveStatus VARCHAR(20),
    ParticipationPeriod VARCHAR(50),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID) ON DELETE SET NULL,
    FOREIGN KEY (ClubID) REFERENCES Club(ClubID) ON DELETE SET NULL
);

INSERT INTO Registration (RegistrationID,StudentID,ClubID,RegistrationDate,ActiveStatus,ParticipationPeriod)
VALUES(201,8230122,1,'2025-03-26','Active','4 Months'),
(202,8230124,2,'2025-03-26','Active','4 Months'),
(203,8230151,3,'2025-03-26','Active','4 Months'),
(204,8230169,4, '2025-03-26','Active','4 Months'),
(205,8230171,5,'2025-03-26','Active','4 Months'),
(206,8230177,6,'2025-03-26','Active','4 Months'),
(207,8230181,7,'2025-03-26','Active','4 Months'),
(208,8230193,8,'2025-03-26','Active','4 Months'),
(209,8230202,9,'2025-03-26','Active','4 Months'),
(210,8230203,10,'2025-03-26','Active','4 Months'),
```

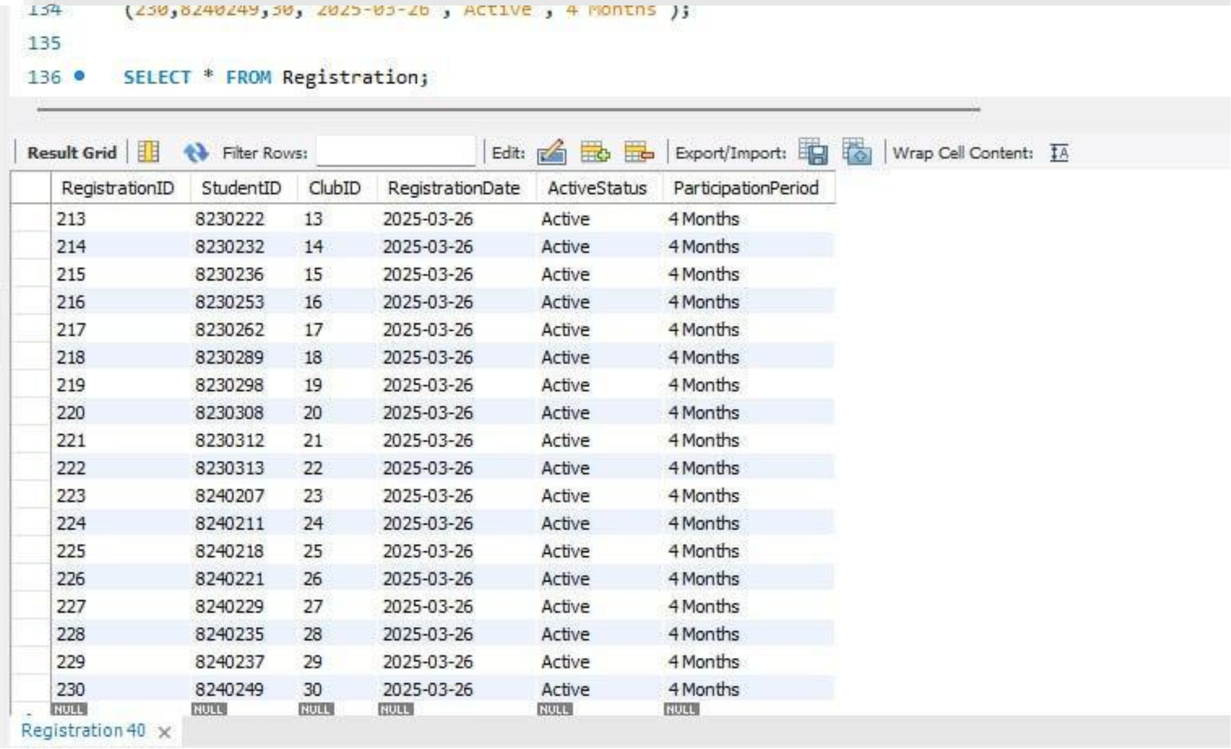


```
(221,8230312,21,'2025-03-26','Active','4 Months'),
(222,8230313,22,'2025-03-26','Active','4 Months'),
(223,8240207,23,'2025-03-26','Active','4 Months'),
(224,8240211,24,'2025-03-26','Active','4 Months'),
(225,8240218,25,'2025-03-26','Active','4 Months'),
(226,8240221,26,'2025-03-26','Active','4 Months'),
(227,8240229,27,'2025-03-26','Active','4 Months'),
(228,8240235,28,'2025-03-26','Active','4 Months'),
(229,8240237,29,'2025-03-26','Active','4 Months'),
(230,8240249,30,'2025-03-26','Active','4 Months');

SELECT * FROM Registration;
```



```
    (229,8240237,29,'2025-03-26','Active','4 Months'),
    (230,8240249,30,'2025-03-26','Active','4 Months');

●   SELECT * FROM Registration;
```

RegistrationID	StudentID	ClubID	RegistrationDate	ActiveStatus	ParticipationPeriod
201	8230122	1	2025-03-26	Active	4 Months
202	8230124	2	2025-03-26	Active	4 Months
203	8230151	3	2025-03-26	Active	4 Months
204	8230169	4	2025-03-26	Active	4 Months
205	8230171	5	2025-03-26	Active	4 Months
206	8230177	6	2025-03-26	Active	4 Months
207	8230181	7	2025-03-26	Active	4 Months
208	8230193	8	2025-03-26	Active	4 Months
209	8230202	9	2025-03-26	Active	4 Months
210	8230203	10	2025-03-26	Active	4 Months
211	8230207	11	2025-03-26	Active	4 Months
212	8230211	12	2025-03-26	Active	4 Months
213	8230222	13	2025-03-26	Active	4 Months
214	8230232	14	2025-03-26	Active	4 Months
215	8230236	15	2025-03-26	Active	4 Months
216	8230253	16	2025-03-26	Active	4 Months
217	8230262	17	2025-03-26	Active	4 Months
218	8230289	18	2025-03-26	Active	4 Months

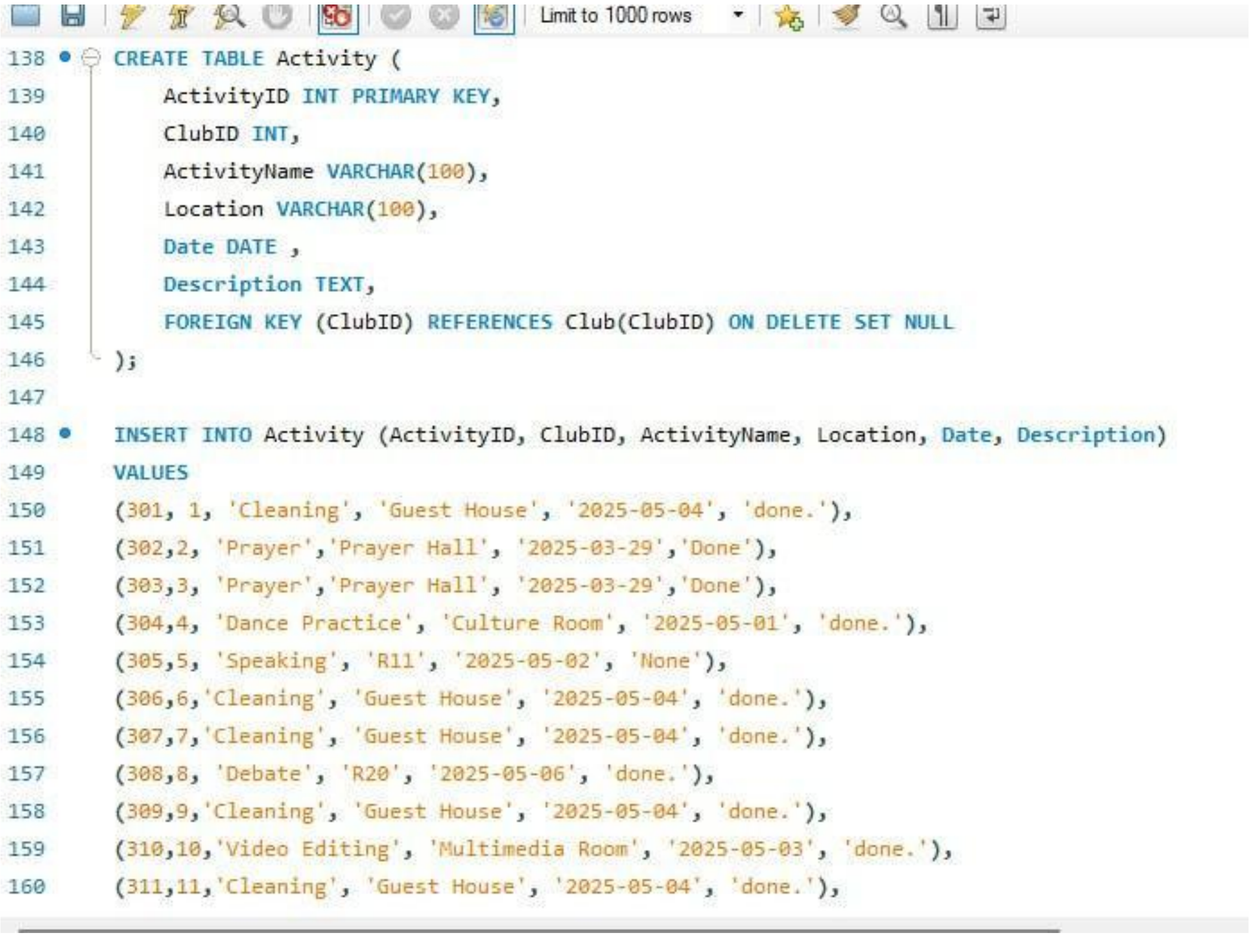


```
    (230,8240249,30,'2025-03-26','Active','4 Months');

●   SELECT * FROM Registration;
```

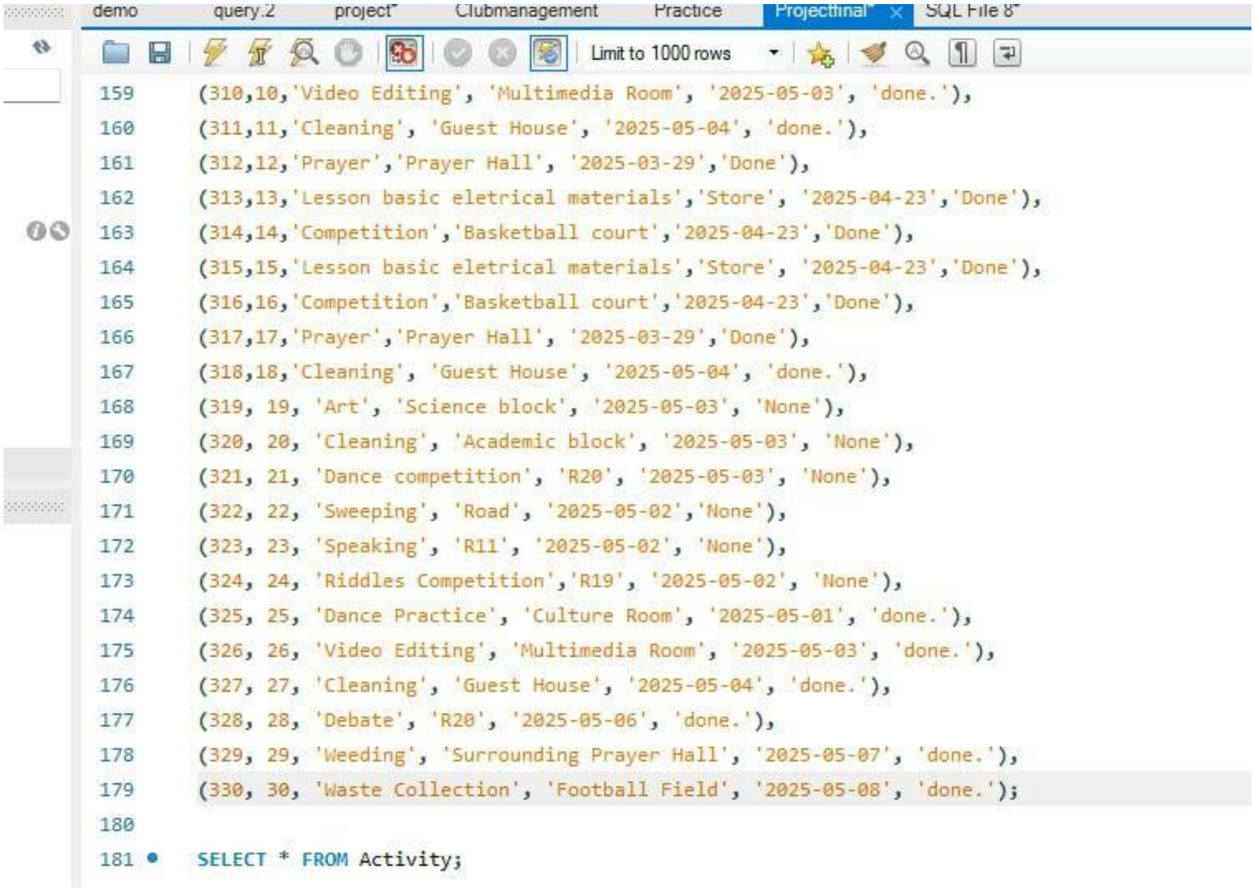
RegistrationID	StudentID	ClubID	RegistrationDate	ActiveStatus	ParticipationPeriod
213	8230222	13	2025-03-26	Active	4 Months
214	8230232	14	2025-03-26	Active	4 Months
215	8230236	15	2025-03-26	Active	4 Months
216	8230253	16	2025-03-26	Active	4 Months
217	8230262	17	2025-03-26	Active	4 Months
218	8230289	18	2025-03-26	Active	4 Months
219	8230298	19	2025-03-26	Active	4 Months
220	8230308	20	2025-03-26	Active	4 Months
221	8230312	21	2025-03-26	Active	4 Months
222	8230313	22	2025-03-26	Active	4 Months
223	8240207	23	2025-03-26	Active	4 Months
224	8240211	24	2025-03-26	Active	4 Months
225	8240218	25	2025-03-26	Active	4 Months
226	8240221	26	2025-03-26	Active	4 Months
227	8240229	27	2025-03-26	Active	4 Months
228	8240235	28	2025-03-26	Active	4 Months
229	8240237	29	2025-03-26	Active	4 Months
230	8240249	30	2025-03-26	Active	4 Months
NULL	NULL	NULL	NULL	NULL	NULL

The Registration table is designed to store information about students' enrollment in various clubs. It includes fields such as RegistrationID, which uniquely identifies each registration record; StudentID and ClubID, which reference the students and clubs involved; RegistrationDate, marking the date when the student joined the club; ActiveStatus, indicating whether the student’s registration is currently active; and ParticipationPeriod, specifying the duration of the student’s involvement. In this dataset, 30 records were inserted, each showing a different student registering for a club on March 26, 2025, with all registrations marked as "Active" and a participation period of four months. When the SELECT * FROM Registration; command is executed, it retrieves all records, displaying a comprehensive list of student memberships in clubs, confirming the data is accurately recorded and available for management or analysis.

5. **Activity table:**



```
CREATE TABLE Activity (
    ActivityID INT PRIMARY KEY,
    ClubID INT,
    ActivityName VARCHAR(100),
    Location VARCHAR(100),
    Date DATE ,
    Description TEXT,
    FOREIGN KEY (ClubID) REFERENCES Club(ClubID) ON DELETE SET NULL
);

INSERT INTO Activity (ActivityID, ClubID, ActivityName, Location, Date, Description)
VALUES
(301, 1, 'Cleaning', 'Guest House', '2025-05-04', 'done.'),
(302,2, 'Prayer','Prayer Hall', '2025-03-29','Done'),
(303,3, 'Prayer','Prayer Hall', '2025-03-29','Done'),
(304,4, 'Dance Practice', 'Culture Room', '2025-05-01', 'done.'),
(305,5, 'Speaking', 'R11', '2025-05-02', 'None'),
(306,6,'Cleaning', 'Guest House', '2025-05-04', 'done.'),
(307,7,'Cleaning', 'Guest House', '2025-05-04', 'done.'),
(308,8, 'Debate', 'R20', '2025-05-06', 'done.'),
(309,9,'Cleaning', 'Guest House', '2025-05-04', 'done.'),
(310,10,'Video Editing', 'Multimedia Room', '2025-05-03', 'done.'),
(311,11,'Cleaning', 'Guest House', '2025-05-04', 'done.'),
```

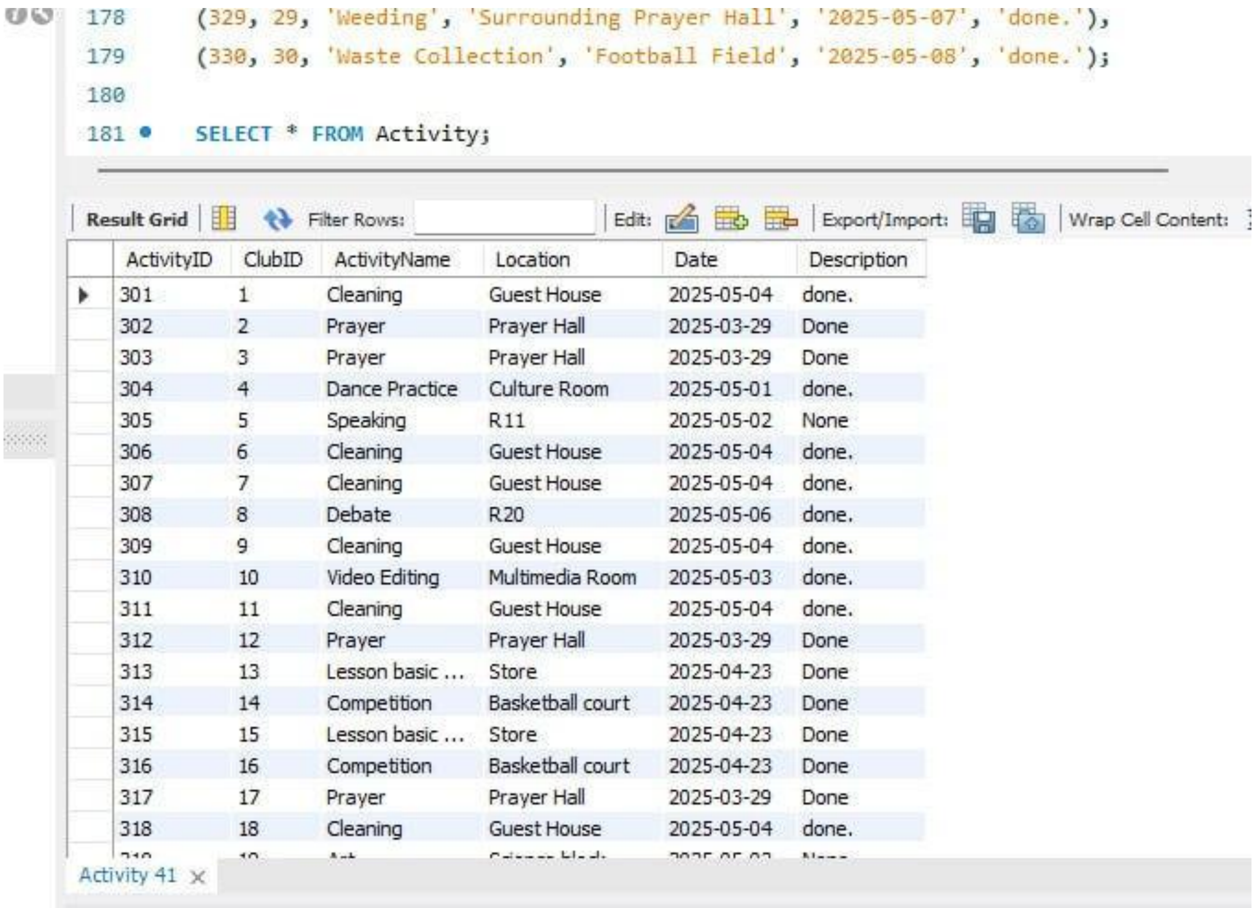


```sql
(310,10,'Video Editing', 'Multimedia Room', '2025-05-03', 'done.'),
(311,11,'Cleaning', 'Guest House', '2025-05-04', 'done.'),
(312,12,'Prayer','Prayer Hall', '2025-03-29','Done'),
(313,13,'Lesson basic eletrical materials','Store', '2025-04-23','Done'),
(314,14,'Competition','Basketball court','2025-04-23','Done'),
(315,15,'Lesson basic eletrical materials','Store', '2025-04-23','Done'),
(316,16,'Competition','Basketball court','2025-04-23','Done'),
(317,17,'Prayer','Prayer Hall', '2025-03-29','Done'),
(318,18,'Cleaning', 'Guest House', '2025-05-04', 'done.'),
(319, 19, 'Art', 'Science block', '2025-05-03', 'None'),
(320, 20, 'Cleaning', 'Academic block', '2025-05-03', 'None'),
(321, 21, 'Dance competition', 'R20', '2025-05-03', 'None'),
(322, 22, 'Sweeping', 'Road', '2025-05-02','None'),
(323, 23, 'Speaking', 'R11', '2025-05-02', 'None'),
(324, 24, 'Riddles Competition','R19', '2025-05-02', 'None'),
(325, 25, 'Dance Practice', 'Culture Room', '2025-05-01', 'done.'),
(326, 26, 'Video Editing', 'Multimedia Room', '2025-05-03', 'done.'),
(327, 27, 'Cleaning', 'Guest House', '2025-05-04', 'done.'),
(328, 28, 'Debate', 'R20', '2025-05-06', 'done.'),
(329, 29, 'Weeding', 'Surrounding Prayer Hall', '2025-05-07', 'done.'),
(330, 30, 'Waste Collection', 'Football Field', '2025-05-08', 'done.');

SELECT * FROM Activity;
```

Explanation

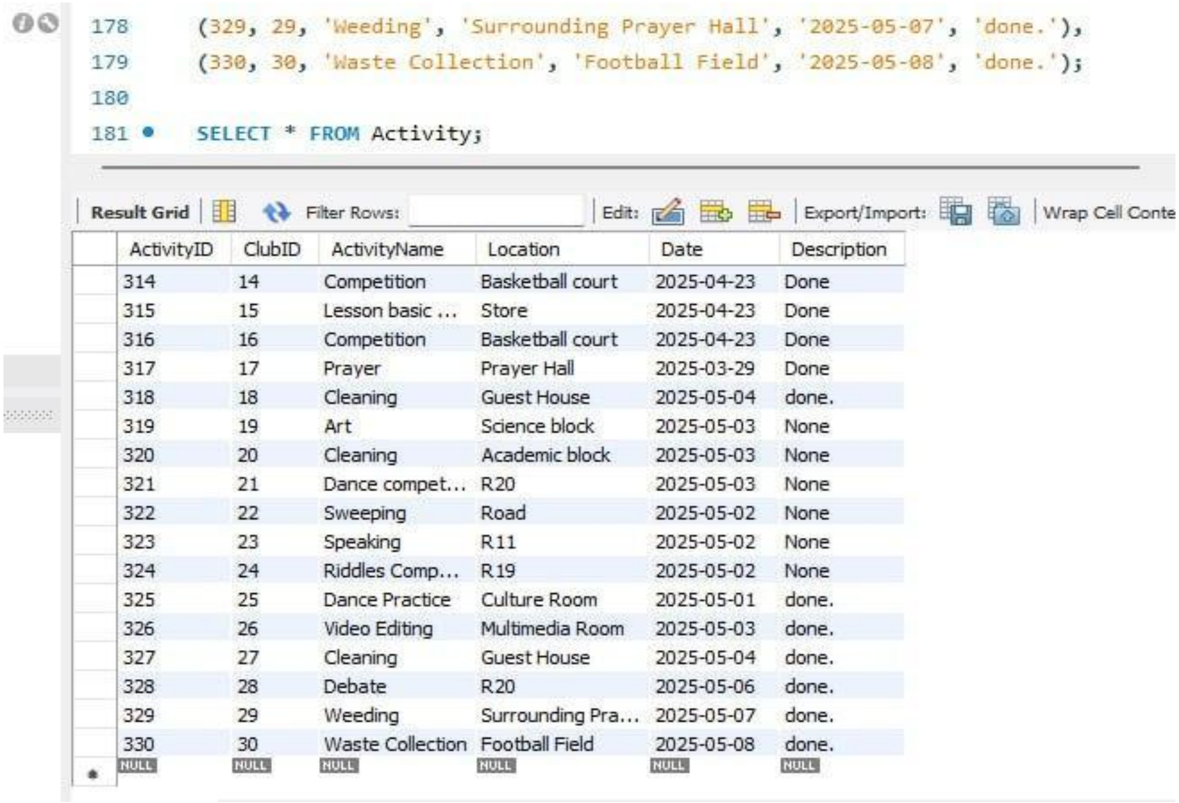
The CREATE TABLE statement sets up the Activity table to store details about events organized by clubs. It includes columns like ActivityID (a unique ID), ClubID (linked to the club), ActivityName, Location, Date, and Description. A foreign key connects ClubID to the Club table, and if a club is deleted, the related ClubID in this table becomes NULL. This design helps manage and track club activities in an organized way.

The INSERT INTO statements adds sample activity records to the Activity table. Each entry includes details like the activity ID, the club organizing it, the event name, location, date, and status. For example, one row shows a cleaning activity by Club 1 at the Guest House on May 4, 2025, marked as "done". These entries help track and manage various events conducted by clubs.



```
(329, 29, 'Weeding', 'Surrounding Prayer Hall', '2025-05-07', 'done.'),
(330, 30, 'Waste Collection', 'Football Field', '2025-05-08', 'done.');

SELECT * FROM Activity;
```

ActivityID	ClubID	ActivityName	Location	Date	Description
301	1	Cleaning	Guest House	2025-05-04	done.
302	2	Prayer	Prayer Hall	2025-03-29	Done
303	3	Prayer	Prayer Hall	2025-03-29	Done
304	4	Dance Practice	Culture Room	2025-05-01	done.
305	5	Speaking	R11	2025-05-02	None
306	6	Cleaning	Guest House	2025-05-04	done.
307	7	Cleaning	Guest House	2025-05-04	done.
308	8	Debate	R20	2025-05-06	done.
309	9	Cleaning	Guest House	2025-05-04	done.
310	10	Video Editing	Multimedia Room	2025-05-03	done.
311	11	Cleaning	Guest House	2025-05-04	done.
312	12	Prayer	Prayer Hall	2025-03-29	Done
313	13	Lesson basic ...	Store	2025-04-23	Done
314	14	Competition	Basketball court	2025-04-23	Done
315	15	Lesson basic ...	Store	2025-04-23	Done
316	16	Competition	Basketball court	2025-04-23	Done
317	17	Prayer	Prayer Hall	2025-03-29	Done
318	18	Cleaning	Guest House	2025-05-04	done.
319	19	Art	Science block	2025-05-02	None



```
(329, 29, 'Weeding', 'Surrounding Prayer Hall', '2025-05-07', 'done.'),
(330, 30, 'Waste Collection', 'Football Field', '2025-05-08', 'done.');

SELECT * FROM Activity;
```

ActivityID	ClubID	ActivityName	Location	Date	Description
314	14	Competition	Basketball court	2025-04-23	Done
315	15	Lesson basic ...	Store	2025-04-23	Done
316	16	Competition	Basketball court	2025-04-23	Done
317	17	Prayer	Prayer Hall	2025-03-29	Done
318	18	Cleaning	Guest House	2025-05-04	done.
319	19	Art	Science block	2025-05-03	None
320	20	Cleaning	Academic block	2025-05-03	None
321	21	Dance compet...	R20	2025-05-03	None
322	22	Sweeping	Road	2025-05-02	None
323	23	Speaking	R11	2025-05-02	None
324	24	Riddles Comp...	R19	2025-05-02	None
325	25	Dance Practice	Culture Room	2025-05-01	done.
326	26	Video Editing	Multimedia Room	2025-05-03	done.
327	27	Cleaning	Guest House	2025-05-04	done.
328	28	Debate	R20	2025-05-06	done.
329	29	Weeding	Surrounding Pra...	2025-05-07	done.
330	30	Waste Collection	Football Field	2025-05-08	done.
NULL	NULL	NULL	NULL	NULL	NULL

Explanation

The image is the result of an SQL query executed on a database table named Activity. The query SELECT * FROM Activity; retrieves all the records and columns from this table. Each row in the table represents a specific activity organized by different clubs. The table includes the following columns:

ActivityID: A unique identifier for each activity.

ClubID: An identifier indicating which club conducted the activity.

ActivityName: The name or type of the activity (e.g., Cleaning, Prayer, Debate).

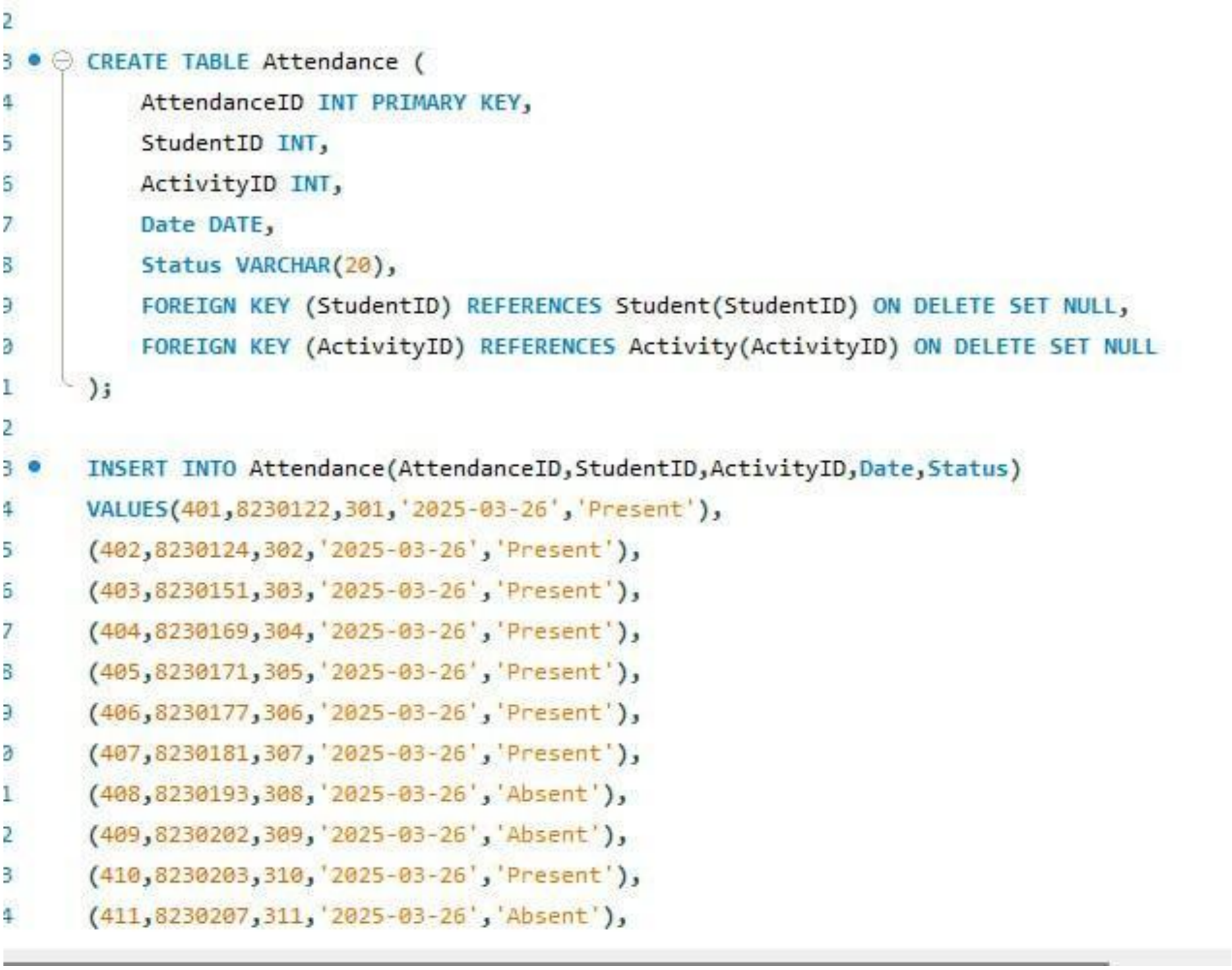
Location: The venue where the activity was held (e.g., Guest House, Prayer Hall).

Date: The date the activity took place (e.g., 2025-05-04).

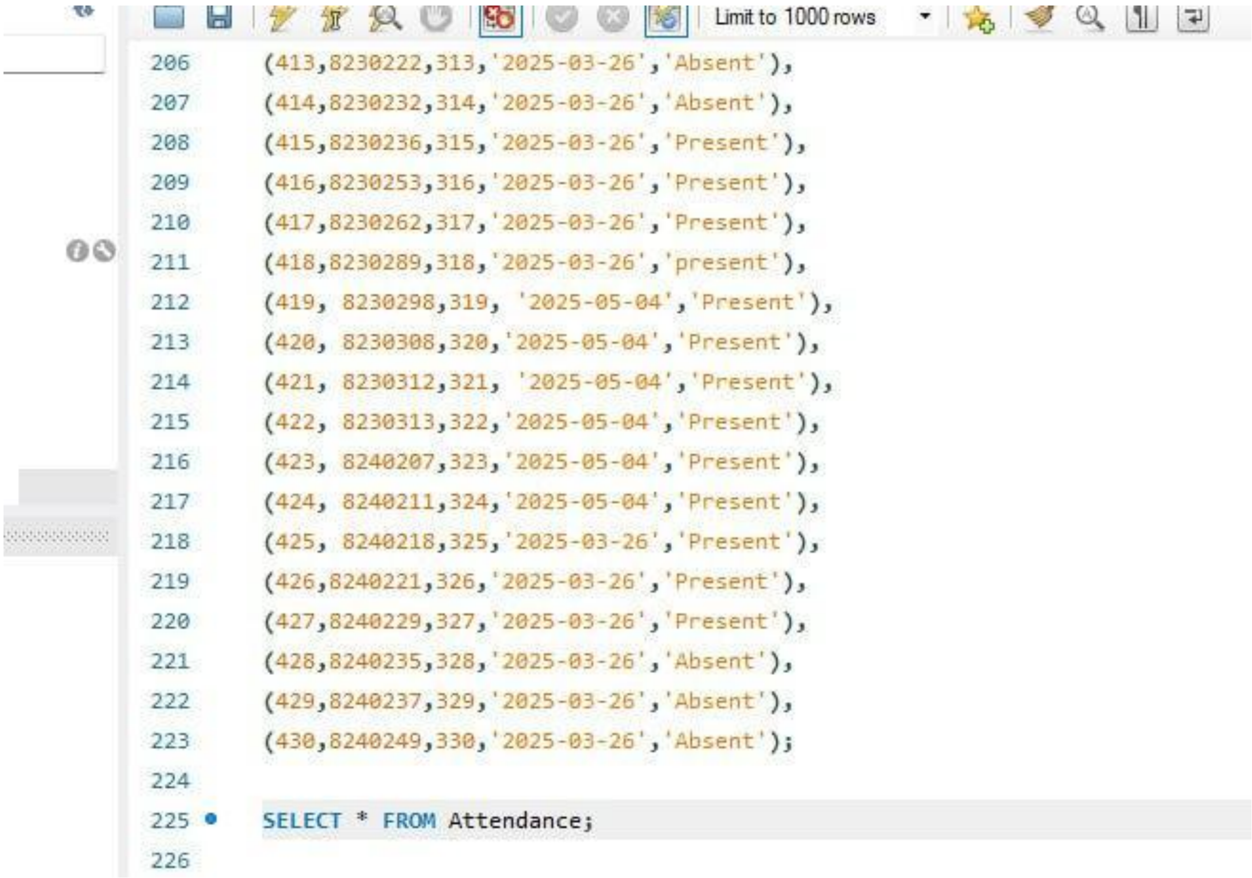
Description: A short status remark for the activity, mostly marked as "Done".

From the data displayed, it is evident that various activities such as cleaning, prayer sessions, speaking, debates, and competitions were conducted at different venues and on different dates, **mostly around** April and May 2025. The "Description" column consistently indicates that these activities have been completed.

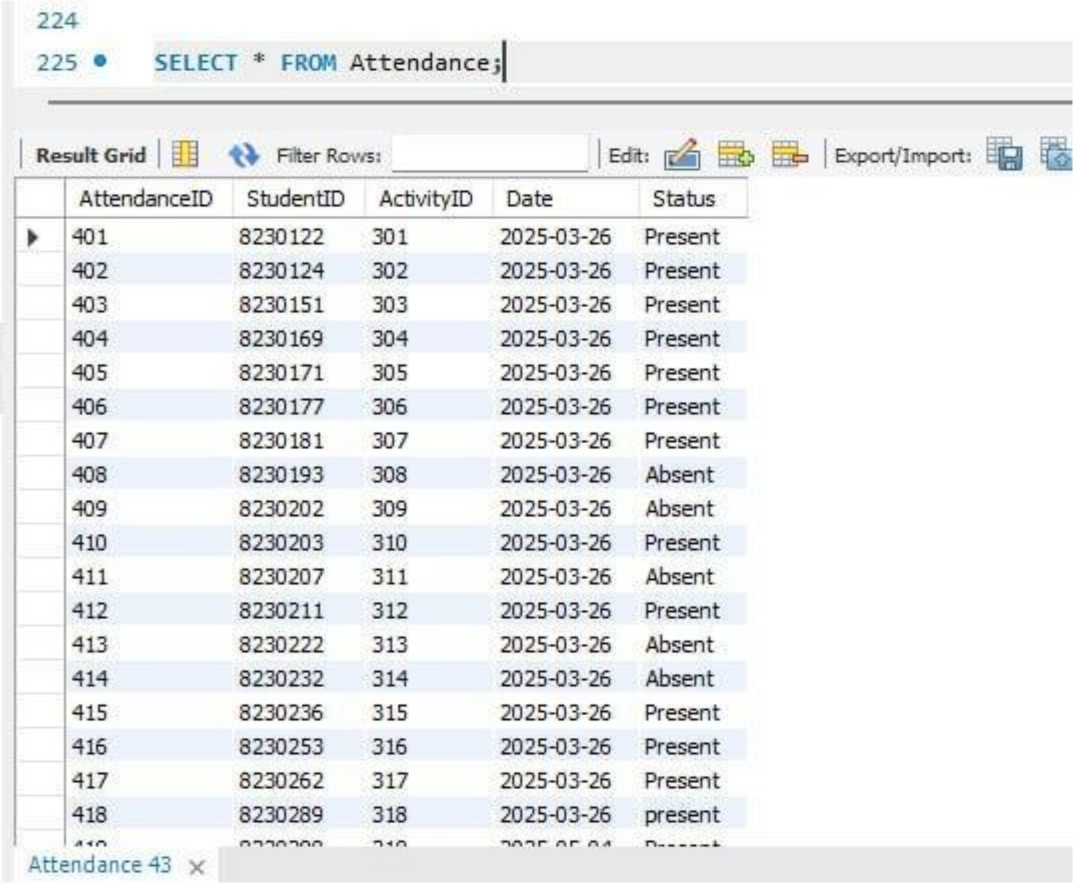
6. Attendance table



```sql

CREATE TABLE Attendance (
    AttendanceID INT PRIMARY KEY,
    StudentID INT,
    ActivityID INT,
    Date DATE,
    Status VARCHAR(20),
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID) ON DELETE SET NULL,
    FOREIGN KEY (ActivityID) REFERENCES Activity(ActivityID) ON DELETE SET NULL
);

INSERT INTO Attendance(AttendanceID,StudentID,ActivityID,Date,Status)
VALUES(401,8230122,301,'2025-03-26','Present'),
(402,8230124,302,'2025-03-26','Present'),
(403,8230151,303,'2025-03-26','Present'),
(404,8230169,304,'2025-03-26','Present'),
(405,8230171,305,'2025-03-26','Present'),
(406,8230177,306,'2025-03-26','Present'),
(407,8230181,307,'2025-03-26','Present'),
(408,8230193,308,'2025-03-26','Absent'),
(409,8230202,309,'2025-03-26','Absent'),
(410,8230203,310,'2025-03-26','Present'),
(411,8230207,311,'2025-03-26','Absent'),
```

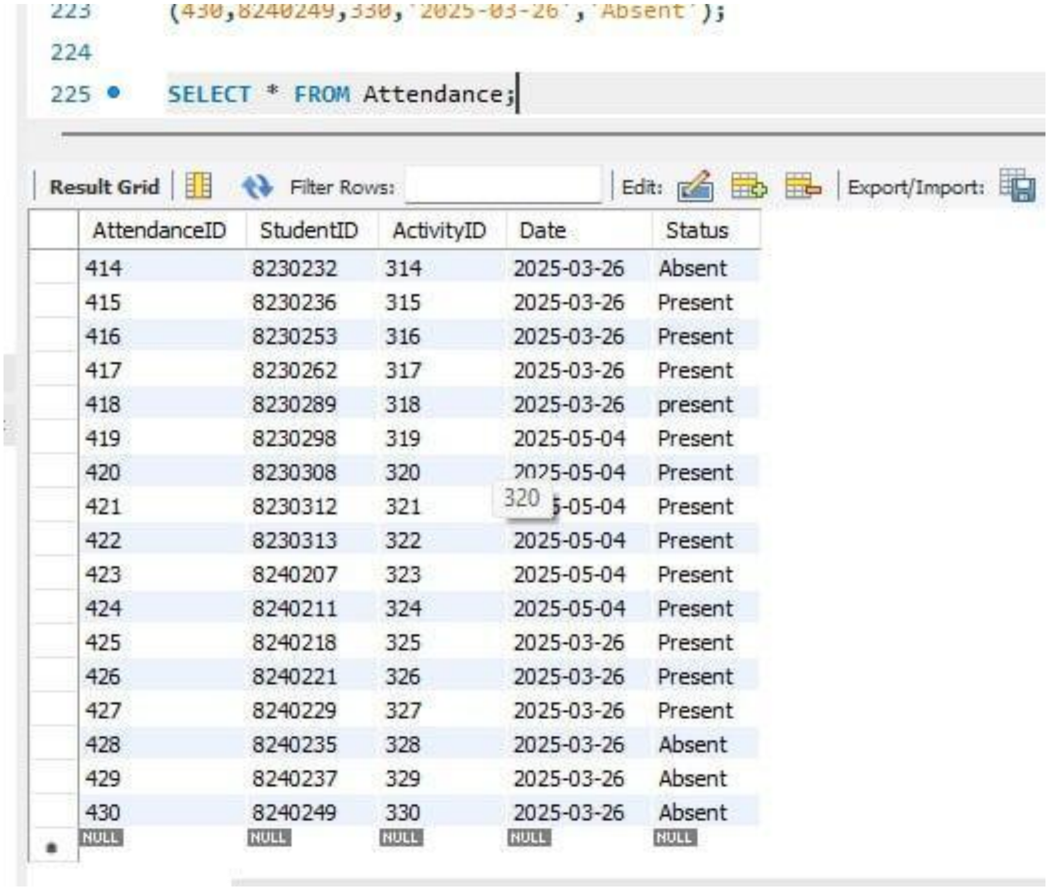


```
(413,8230222,313,'2025-03-26','Absent'),
(414,8230232,314,'2025-03-26','Absent'),
(415,8230236,315,'2025-03-26','Present'),
(416,8230253,316,'2025-03-26','Present'),
(417,8230262,317,'2025-03-26','Present'),
(418,8230289,318,'2025-03-26','present'),
(419, 8230298,319, '2025-05-04','Present'),
(420, 8230308,320,'2025-05-04','Present'),
(421, 8230312,321, '2025-05-04','Present'),
(422, 8230313,322,'2025-05-04','Present'),
(423, 8240207,323,'2025-05-04','Present'),
(424, 8240211,324,'2025-05-04','Present'),
(425, 8240218,325,'2025-03-26','Present'),
(426,8240221,326,'2025-03-26','Present'),
(427,8240229,327,'2025-03-26','Present'),
(428,8240235,328,'2025-03-26','Absent'),
(429,8240237,329,'2025-03-26','Absent'),
(430,8240249,330,'2025-03-26','Absent');

SELECT * FROM Attendance;

```



```sql
SELECT * FROM Attendance;
```

AttendanceID	StudentID	ActivityID	Date	Status
401	8230122	301	2025-03-26	Present
402	8230124	302	2025-03-26	Present
403	8230151	303	2025-03-26	Present
404	8230169	304	2025-03-26	Present
405	8230171	305	2025-03-26	Present
406	8230177	306	2025-03-26	Present
407	8230181	307	2025-03-26	Present
408	8230193	308	2025-03-26	Absent
409	8230202	309	2025-03-26	Absent
410	8230203	310	2025-03-26	Present
411	8230207	311	2025-03-26	Absent
412	8230211	312	2025-03-26	Present
413	8230222	313	2025-03-26	Absent
414	8230232	314	2025-03-26	Absent
415	8230236	315	2025-03-26	Present
416	8230253	316	2025-03-26	Present
417	8230262	317	2025-03-26	Present
418	8230289	318	2025-03-26	present

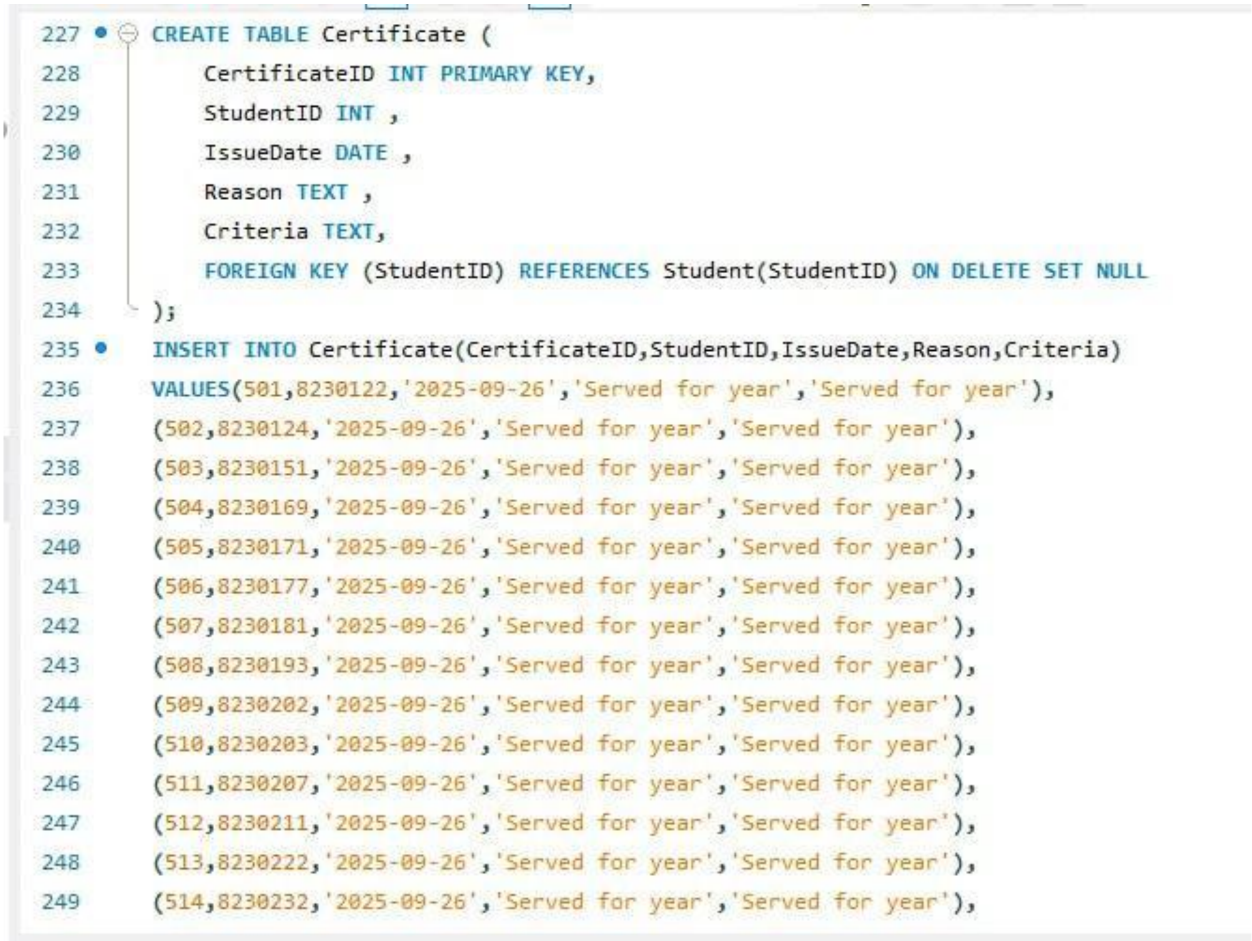


```
    (430,8240249,330,'2025-03-26','Absent');

    SELECT * FROM Attendance;
```

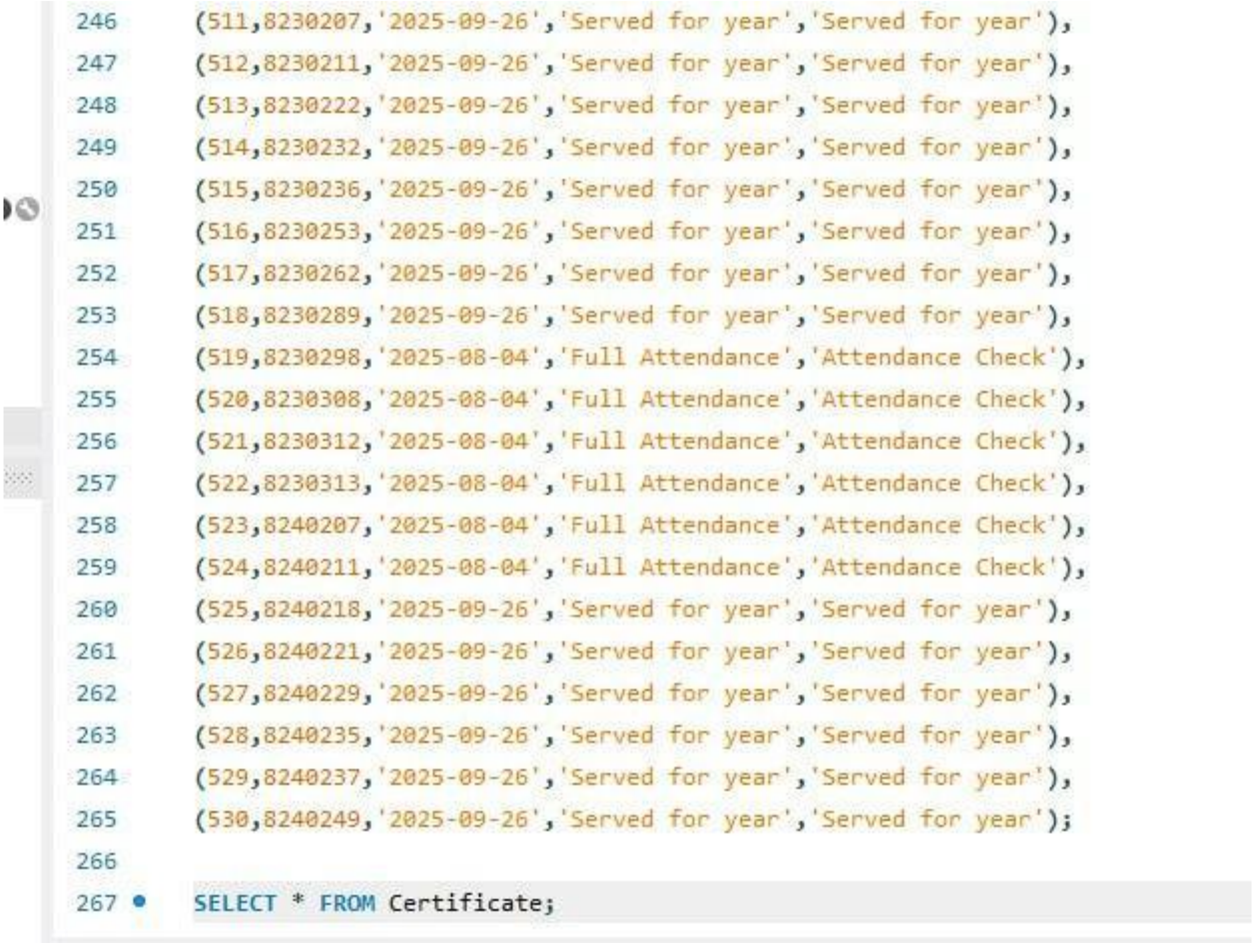
AttendanceID	StudentID	ActivityID	Date	Status
414	8230232	314	2025-03-26	Absent
415	8230236	315	2025-03-26	Present
416	8230253	316	2025-03-26	Present
417	8230262	317	2025-03-26	Present
418	8230289	318	2025-03-26	present
419	8230298	319	2025-05-04	Present
420	8230308	320	2025-05-04	Present
421	8230312	321	5-05-04	Present
422	8230313	322	2025-05-04	Present
423	8240207	323	2025-05-04	Present
424	8240211	324	2025-05-04	Present
425	8240218	325	2025-03-26	Present
426	8240221	326	2025-03-26	Present
427	8240229	327	2025-03-26	Present
428	8240235	328	2025-03-26	Absent
429	8240237	329	2025-03-26	Absent
430	8240249	330	2025-03-26	Absent
NULL	NULL	NULL	NULL	NULL

The Attendance table is designed to store information about students' participation in various activities. Each record in the table is uniquely identified by the AttendanceID. The table includes StudentID and ActivityID, which link each attendance entry to a specific student and activity through foreign key relationships. These foreign keys are configured with ON DELETE SET NULL, ensuring that if a student or activity is removed from the system, the related attendance record remains intact with the corresponding field set to null. The Date field records the specific day the attendance was taken, while the Status field indicates whether the student was "Present" or "Absent" on that date. In this dataset, six attendance records were added for March 26, 2025. Three students were marked as present and three as absent for different activities. When the SELECT * FROM Attendance; command is executed, it retrieves all the records from the table, offering a detailed view of student attendance that can be used for monitoring participation and supporting administrative decisions.

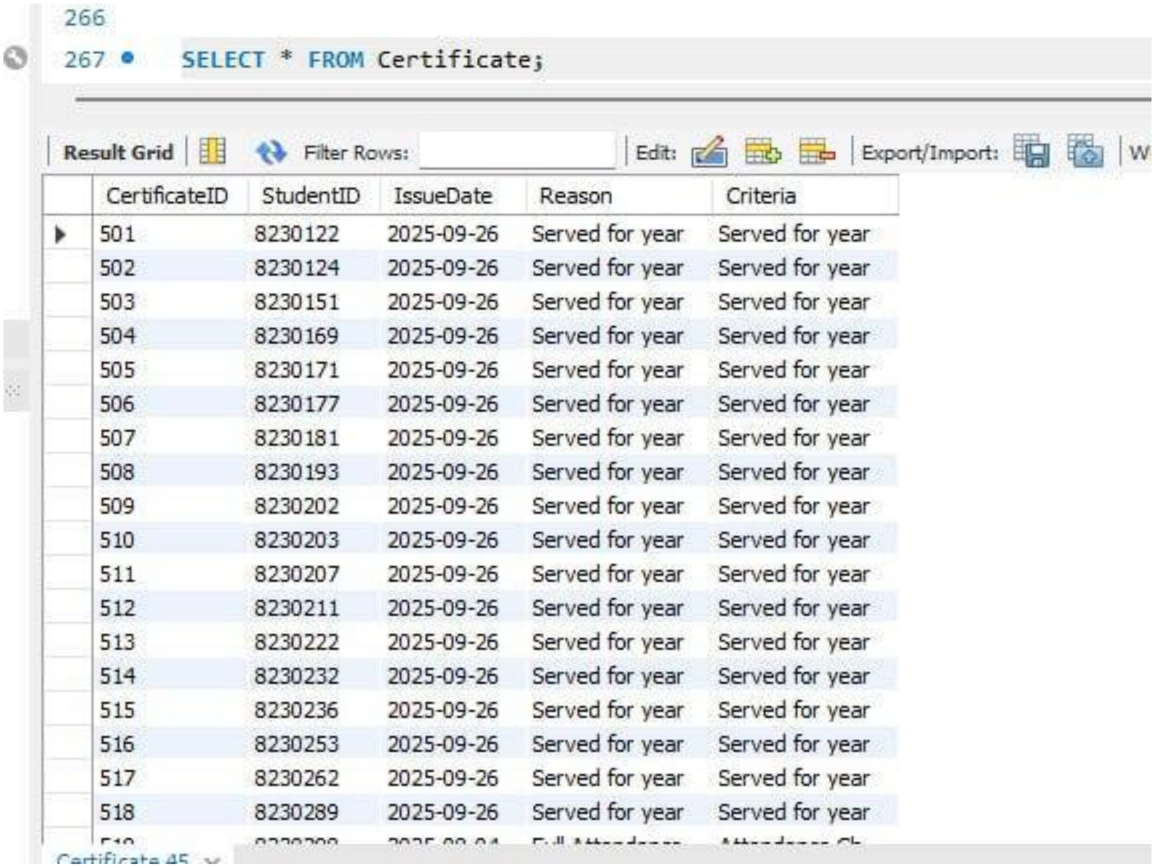
7. Certificate table



```sql
CREATE TABLE Certificate (
    CertificateID INT PRIMARY KEY,
    StudentID INT ,
    IssueDate DATE ,
    Reason TEXT ,
    Criteria TEXT,
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID) ON DELETE SET NULL
);
INSERT INTO Certificate(CertificateID,StudentID,IssueDate,Reason,Criteria)
VALUES(501,8230122,'2025-09-26','Served for year','Served for year'),
(502,8230124,'2025-09-26','Served for year','Served for year'),
(503,8230151,'2025-09-26','Served for year','Served for year'),
(504,8230169,'2025-09-26','Served for year','Served for year'),
(505,8230171,'2025-09-26','Served for year','Served for year'),
(506,8230177,'2025-09-26','Served for year','Served for year'),
(507,8230181,'2025-09-26','Served for year','Served for year'),
(508,8230193,'2025-09-26','Served for year','Served for year'),
(509,8230202,'2025-09-26','Served for year','Served for year'),
(510,8230203,'2025-09-26','Served for year','Served for year'),
(511,8230207,'2025-09-26','Served for year','Served for year'),
(512,8230211,'2025-09-26','Served for year','Served for year'),
(513,8230222,'2025-09-26','Served for year','Served for year'),
(514,8230232,'2025-09-26','Served for year','Served for year'),
```

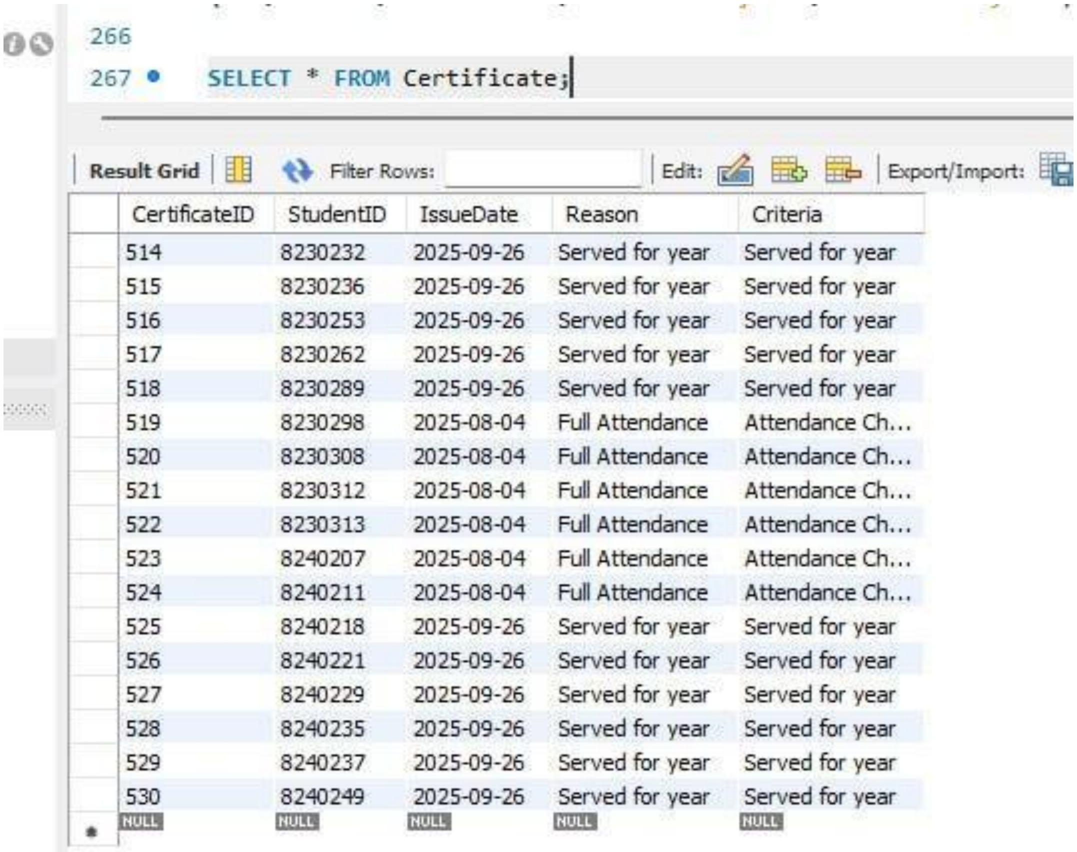


```
(511,8230207,'2025-09-26','Served for year','Served for year'),
(512,8230211,'2025-09-26','Served for year','Served for year'),
(513,8230222,'2025-09-26','Served for year','Served for year'),
(514,8230232,'2025-09-26','Served for year','Served for year'),
(515,8230236,'2025-09-26','Served for year','Served for year'),
(516,8230253,'2025-09-26','Served for year','Served for year'),
(517,8230262,'2025-09-26','Served for year','Served for year'),
(518,8230289,'2025-09-26','Served for year','Served for year'),
(519,8230298,'2025-08-04','Full Attendance','Attendance Check'),
(520,8230308,'2025-08-04','Full Attendance','Attendance Check'),
(521,8230312,'2025-08-04','Full Attendance','Attendance Check'),
(522,8230313,'2025-08-04','Full Attendance','Attendance Check'),
(523,8240207,'2025-08-04','Full Attendance','Attendance Check'),
(524,8240211,'2025-08-04','Full Attendance','Attendance Check'),
(525,8240218,'2025-09-26','Served for year','Served for year'),
(526,8240221,'2025-09-26','Served for year','Served for year'),
(527,8240229,'2025-09-26','Served for year','Served for year'),
(528,8240235,'2025-09-26','Served for year','Served for year'),
(529,8240237,'2025-09-26','Served for year','Served for year'),
(530,8240249,'2025-09-26','Served for year','Served for year');

SELECT * FROM Certificate;
```



```
SELECT * FROM Certificate;
```

CertificateID	StudentID	IssueDate	Reason	Criteria
501	8230122	2025-09-26	Served for year	Served for year
502	8230124	2025-09-26	Served for year	Served for year
503	8230151	2025-09-26	Served for year	Served for year
504	8230169	2025-09-26	Served for year	Served for year
505	8230171	2025-09-26	Served for year	Served for year
506	8230177	2025-09-26	Served for year	Served for year
507	8230181	2025-09-26	Served for year	Served for year
508	8230193	2025-09-26	Served for year	Served for year
509	8230202	2025-09-26	Served for year	Served for year
510	8230203	2025-09-26	Served for year	Served for year
511	8230207	2025-09-26	Served for year	Served for year
512	8230211	2025-09-26	Served for year	Served for year
513	8230222	2025-09-26	Served for year	Served for year
514	8230232	2025-09-26	Served for year	Served for year
515	8230236	2025-09-26	Served for year	Served for year
516	8230253	2025-09-26	Served for year	Served for year
517	8230262	2025-09-26	Served for year	Served for year
518	8230289	2025-09-26	Served for year	Served for year
519	8230299	2025-09-04	Full Attendance	Attendance Ch



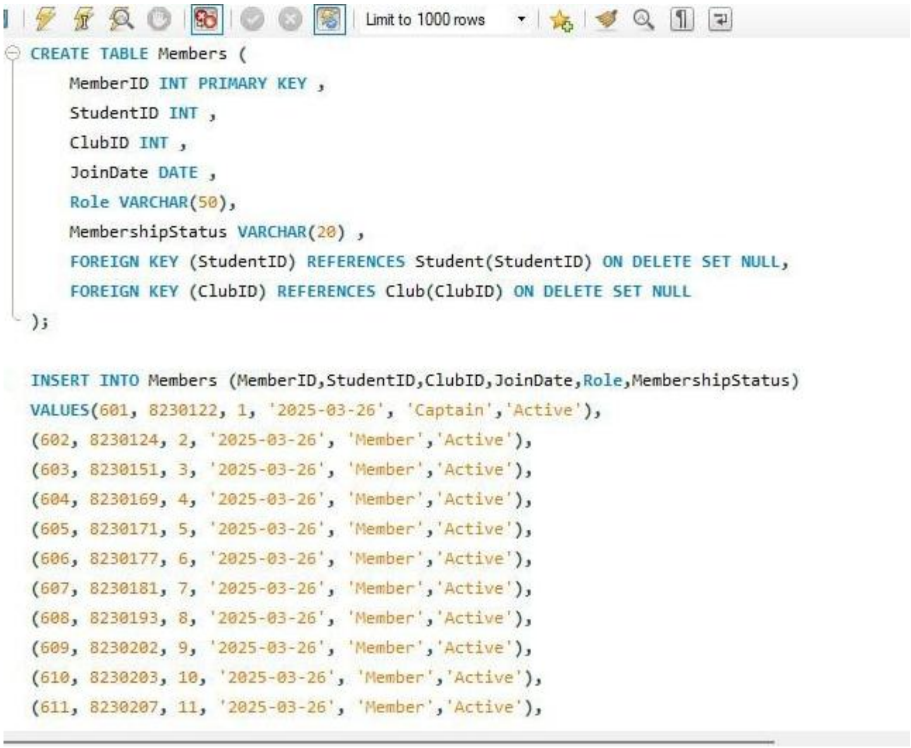
```
SELECT * FROM Certificate;
```

CertificateID	StudentID	IssueDate	Reason	Criteria
514	8230232	2025-09-26	Served for year	Served for year
515	8230236	2025-09-26	Served for year	Served for year
516	8230253	2025-09-26	Served for year	Served for year
517	8230262	2025-09-26	Served for year	Served for year
518	8230289	2025-09-26	Served for year	Served for year
519	8230298	2025-08-04	Full Attendance	Attendance Ch...
520	8230308	2025-08-04	Full Attendance	Attendance Ch...
521	8230312	2025-08-04	Full Attendance	Attendance Ch...
522	8230313	2025-08-04	Full Attendance	Attendance Ch...
523	8240207	2025-08-04	Full Attendance	Attendance Ch...
524	8240211	2025-08-04	Full Attendance	Attendance Ch...
525	8240218	2025-09-26	Served for year	Served for year
526	8240221	2025-09-26	Served for year	Served for year
527	8240229	2025-09-26	Served for year	Served for year
528	8240235	2025-09-26	Served for year	Served for year
529	8240237	2025-09-26	Served for year	Served for year
530	8240249	2025-09-26	Served for year	Served for year
NULL	NULL	NULL	NULL	NULL

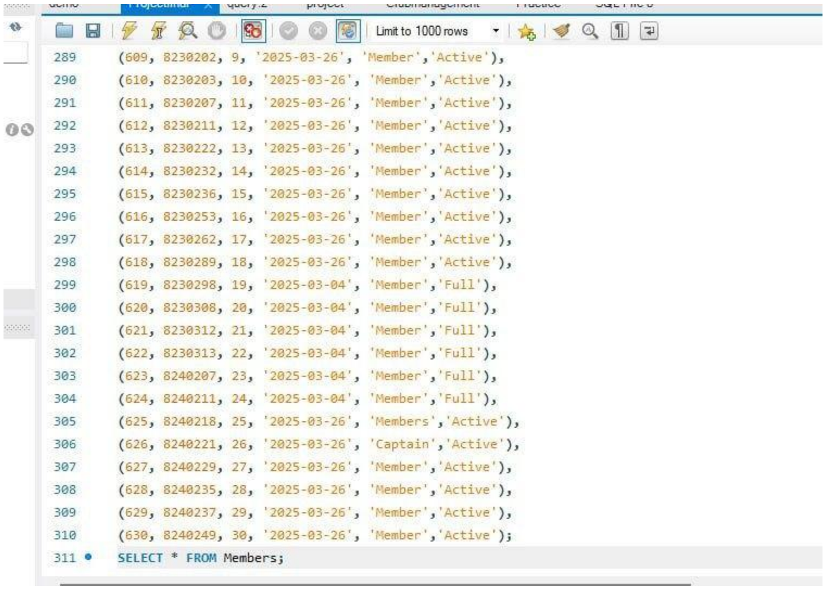
The Certificate table is created to store records of certificates issued to students in recognition of their participation or contribution. Each record is uniquely identified by the CertificateID. The table includes a StudentID, which links each certificate to a specific student through a foreign key relationship. This relationship is designed with ON DELETE SET NULL, meaning that if a student's record is deleted, the corresponding student reference in the certificate will be set to null, preserving the certificate record itself. The IssueDate field indicates the date on which the certificate was awarded, while the Reason and Criteria fields describe why the certificate was issued and the basis on which it was granted.In this dataset, six certificates were issued on September 26, 2025, to six different students. All certificates were awarded for the same reason and based on the same criteria "Served for year" acknowledging their year-long service. When the command SELECT * FROM Certificate; is executed, retrieves all the entries in the table, providing a clear overview of student recognitions and the justification for each award. This data can be used for tracking student achievements and maintaining institutional records of participation.

8. Members table:

- **First step:** creating a table called members to store the information about members of the club. And inserting the information and populated.



```
CREATE TABLE Members (
    MemberID INT PRIMARY KEY ,
    StudentID INT ,
    ClubID INT ,
    JoinDate DATE ,
    Role VARCHAR(50),
    MembershipStatus VARCHAR(20) ,
    FOREIGN KEY (StudentID) REFERENCES Student(StudentID) ON DELETE SET NULL,
    FOREIGN KEY (ClubID) REFERENCES Club(ClubID) ON DELETE SET NULL
);

INSERT INTO Members (MemberID,StudentID,ClubID,JoinDate,Role,MembershipStatus)
VALUES(601, 8230122, 1, '2025-03-26', 'Captain','Active'),
(602, 8230124, 2, '2025-03-26', 'Member','Active'),
(603, 8230151, 3, '2025-03-26', 'Member','Active'),
(604, 8230169, 4, '2025-03-26', 'Member','Active'),
(605, 8230171, 5, '2025-03-26', 'Member','Active'),
(606, 8230177, 6, '2025-03-26', 'Member','Active'),
(607, 8230181, 7, '2025-03-26', 'Member','Active'),
(608, 8230193, 8, '2025-03-26', 'Member','Active'),
(609, 8230202, 9, '2025-03-26', 'Member','Active'),
(610, 8230203, 10, '2025-03-26', 'Member','Active'),
(611, 8230207, 11, '2025-03-26', 'Member','Active'),
```

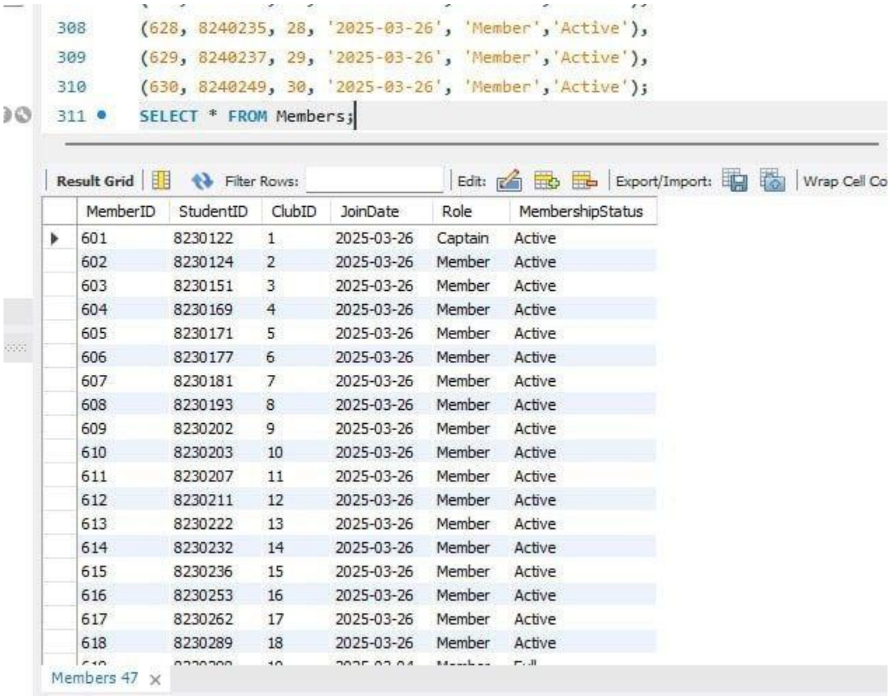


```
    (609, 8230202, 9, '2025-03-26', 'Member','Active'),
    (610, 8230203, 10, '2025-03-26', 'Member','Active'),
    (611, 8230207, 11, '2025-03-26', 'Member','Active'),
    (612, 8230211, 12, '2025-03-26', 'Member','Active'),
    (613, 8230222, 13, '2025-03-26', 'Member','Active'),
    (614, 8230232, 14, '2025-03-26', 'Member','Active'),
    (615, 8230236, 15, '2025-03-26', 'Member','Active'),
    (616, 8230253, 16, '2025-03-26', 'Member','Active'),
    (617, 8230262, 17, '2025-03-26', 'Member','Active'),
    (618, 8230289, 18, '2025-03-26', 'Member','Active'),
    (619, 8230298, 19, '2025-03-04', 'Member','Full'),
    (620, 8230308, 20, '2025-03-04', 'Member','Full'),
    (621, 8230312, 21, '2025-03-04', 'Member','Full'),
    (622, 8230313, 22, '2025-03-04', 'Member','Full'),
    (623, 8240207, 23, '2025-03-04', 'Member','Full'),
    (624, 8240211, 24, '2025-03-04', 'Member','Full'),
    (625, 8240218, 25, '2025-03-26', 'Members','Active'),
    (626, 8240221, 26, '2025-03-26', 'Captain','Active'),
    (627, 8240229, 27, '2025-03-26', 'Member','Active'),
    (628, 8240235, 28, '2025-03-26', 'Member','Active'),
    (629, 8240237, 29, '2025-03-26', 'Member','Active'),
    (630, 8240249, 30, '2025-03-26', 'Member','Active');
    SELECT * FROM Members;
```

Explanation

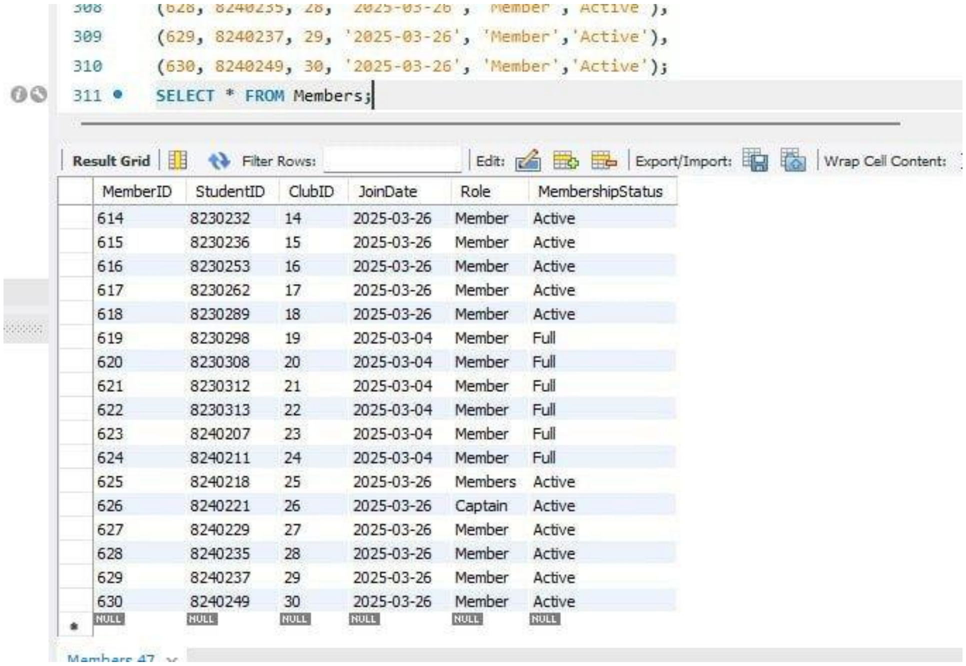
The SQL script creates a Members table to store student club membership details, including member ID, student ID, club ID, join date, role, and membership status. It establishes foreign key relationships with Student and Club tables and inserts sample data where all members joined on '2025-03-26' with roles like 'Captain' or 'Member' and have an 'Active' status.

The INSERT INTO statement adds multiple members to the Members table with their IDs, student and club IDs, join date ('2025-03-26'), roles ('Captain' or 'Member'), and status ('Active').



```
    (628, 8240235, 28, '2025-03-26', 'Member','Active'),
    (629, 8240237, 29, '2025-03-26', 'Member','Active'),
    (630, 8240249, 30, '2025-03-26', 'Member','Active');
SELECT * FROM Members;
```

MemberID	StudentID	ClubID	JoinDate	Role	MembershipStatus
601	8230122	1	2025-03-26	Captain	Active
602	8230124	2	2025-03-26	Member	Active
603	8230151	3	2025-03-26	Member	Active
604	8230169	4	2025-03-26	Member	Active
605	8230171	5	2025-03-26	Member	Active
606	8230177	6	2025-03-26	Member	Active
607	8230181	7	2025-03-26	Member	Active
608	8230193	8	2025-03-26	Member	Active
609	8230202	9	2025-03-26	Member	Active
610	8230203	10	2025-03-26	Member	Active
611	8230207	11	2025-03-26	Member	Active
612	8230211	12	2025-03-26	Member	Active
613	8230222	13	2025-03-26	Member	Active
614	8230232	14	2025-03-26	Member	Active
615	8230236	15	2025-03-26	Member	Active
616	8230253	16	2025-03-26	Member	Active
617	8230262	17	2025-03-26	Member	Active
618	8230289	18	2025-03-26	Member	Active

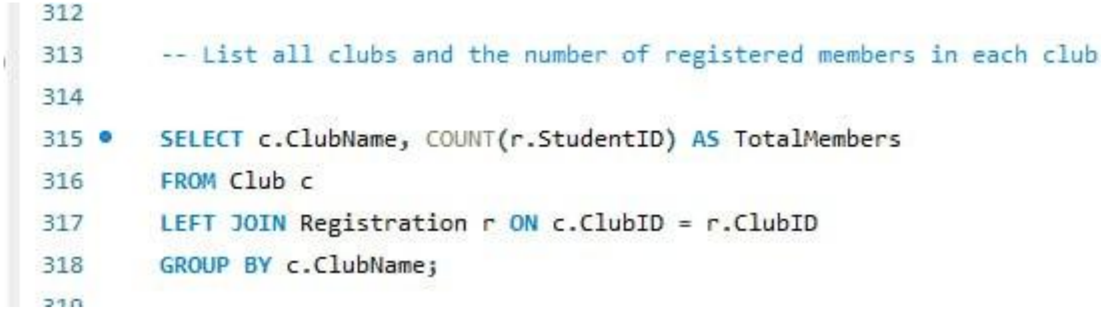


```
(628, 8240235, 28, '2025-03-26', 'Member','Active'),
(629, 8240237, 29, '2025-03-26', 'Member','Active'),
(630, 8240249, 30, '2025-03-26', 'Member','Active');
SELECT * FROM Members;
```

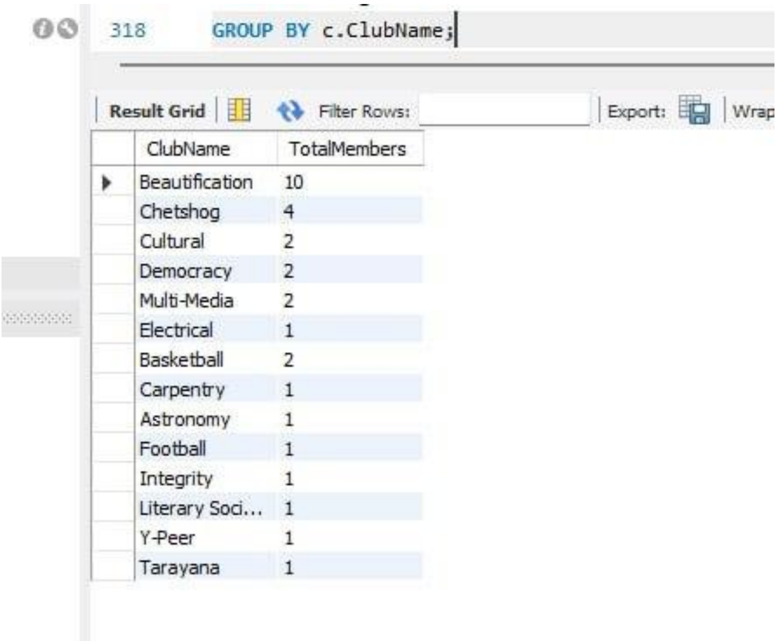
MemberID	StudentID	ClubID	JoinDate	Role	MembershipStatus
614	8230232	14	2025-03-26	Member	Active
615	8230236	15	2025-03-26	Member	Active
616	8230253	16	2025-03-26	Member	Active
617	8230262	17	2025-03-26	Member	Active
618	8230289	18	2025-03-26	Member	Active
619	8230298	19	2025-03-04	Member	Full
620	8230308	20	2025-03-04	Member	Full
621	8230312	21	2025-03-04	Member	Full
622	8230313	22	2025-03-04	Member	Full
623	8240207	23	2025-03-04	Member	Full
624	8240211	24	2025-03-04	Member	Full
625	8240218	25	2025-03-26	Members	Active
626	8240221	26	2025-03-26	Captain	Active
627	8240229	27	2025-03-26	Member	Active
628	8240235	28	2025-03-26	Member	Active
629	8240237	29	2025-03-26	Member	Active
630	8240249	30	2025-03-26	Member	Active
NULL	NULL	NULL	NULL	NULL	NULL

The image shows the result of the SQL query SELECT * FROM Members;, which retrieves all records from the Members table. It displays details like MemberID, StudentID, ClubID, JoinDate, Role, and MembershipStatus. All members joined on '2025-03-26', have the role 'Member' or 'Captain', and are marked as 'Active'.

9. Listing all clubs and the number of registered member in each club



```

-- List all clubs and the number of registered members in each club

SELECT c.ClubName, COUNT(r.StudentID) AS TotalMembers
FROM Club c
LEFT JOIN Registration r ON c.ClubID = r.ClubID
GROUP BY c.ClubName;
```



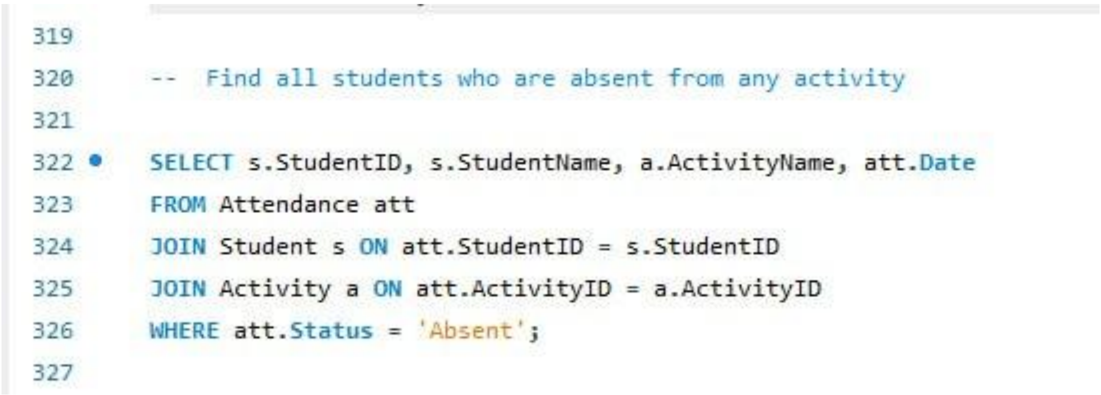
```
GROUP BY c.ClubName;
```

ClubName	TotalMembers
Beautification	10
Chetshog	4
Cultural	2
Democracy	2
Multi-Media	2
Electrical	1
Basketball	2
Carpentry	1
Astronomy	1
Football	1
Integrity	1
Literary Soci...	1
Y-Peer	1
Tarayana	1

Explanation

The above SQL query is designed to retrieve a list of students along with the names of the clubs they are registered in. It starts by selecting the student name from the Student table and the club name from the Club table. The Student table is aliased as s, the Registration table as r, and the Club table as c. The query first joins the Student and Registration tables using the StudentID to match each student with their registration records. Then, it joins the Registration and Club tables using the ClubID to link each registration to the respective club. As a result, the query outputs a list showing each student's name alongside the name of the club they are enrolled in.

10. Finding all the students who are absent for any activity

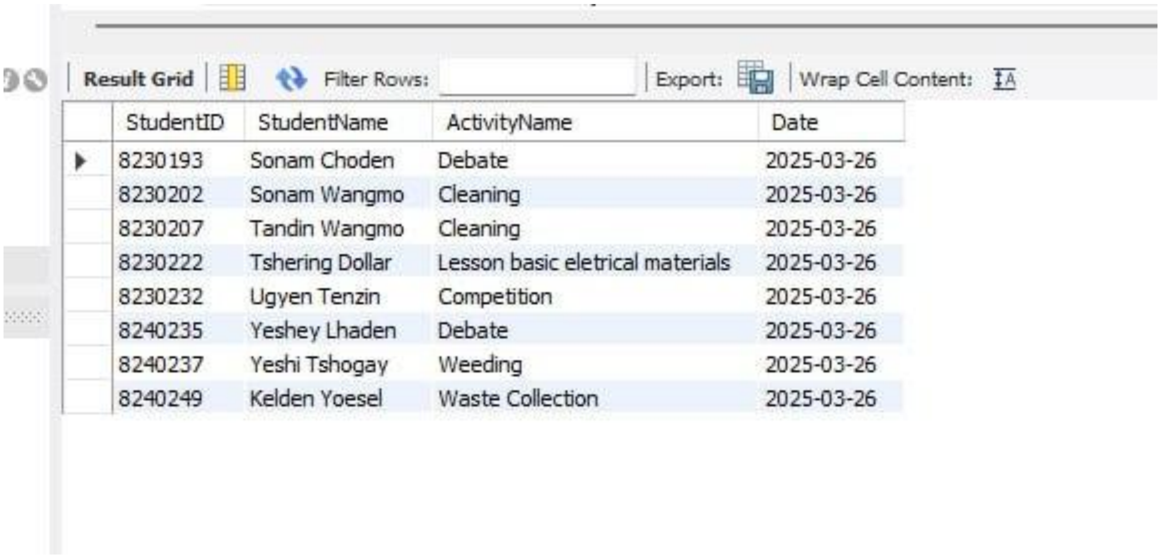


```

--  Find all students who are absent from any activity

SELECT s.StudentID, s.StudentName, a.ActivityName, att.Date
FROM Attendance att
JOIN Student s ON att.StudentID = s.StudentID
JOIN Activity a ON att.ActivityID = a.ActivityID
WHERE att.Status = 'Absent';

```

Explanation

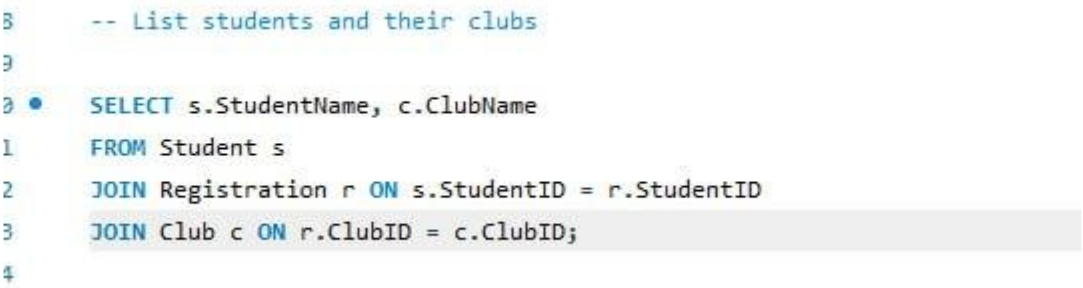
This SQL query is designed to retrieve a list of students who were absent from any activity. It begins by selecting data from the Attendance table, which records student attendance for various activities, and assigns it an alias att for easier reference. The query then performs two inner joins: one with the Student table to match each attendance record with the corresponding student's information, and another with the Activity table to associate each record with the relevant activity details. The WHERE clause filters the results to include only those records where the attendance status is marked as 'Absent'.

Finally, the SELECT clause outputs the student ID, student name, activity name, and the date of absence. The result is a clear and useful list of students who have been absent from one or more activities, along with the specific activity and date, which can be used for tracking attendance and analyzing participation.

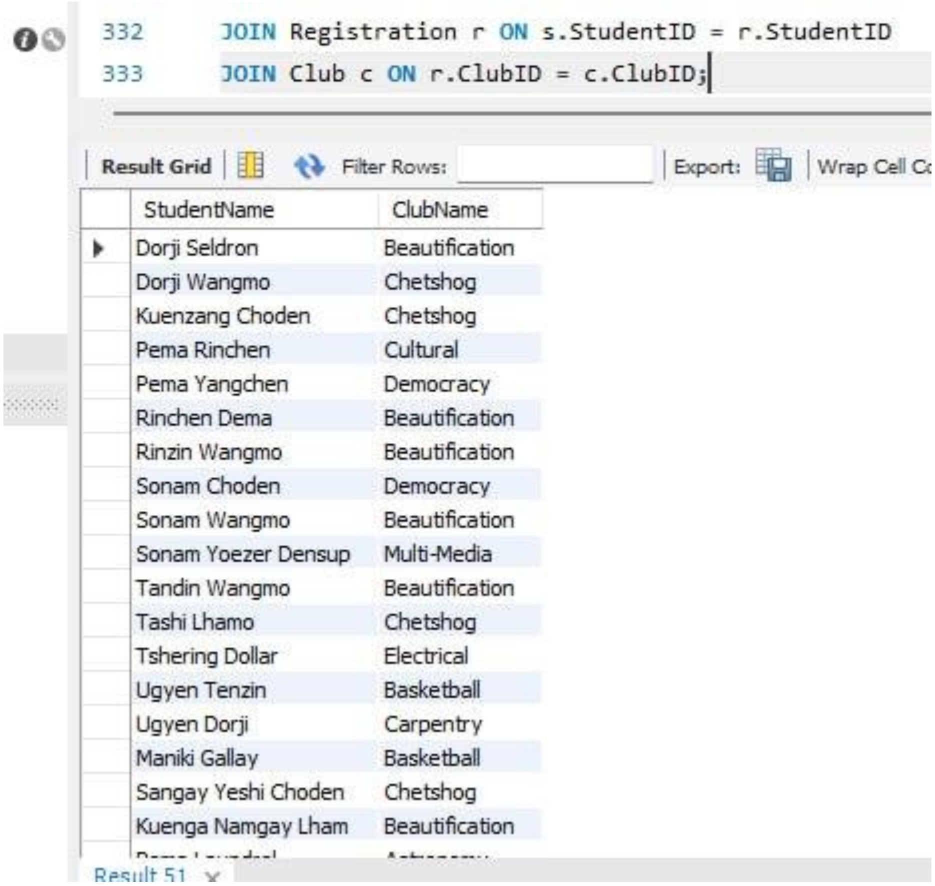


StudentID	StudentName	ActivityName	Date
8230193	Sonam Choden	Debate	2025-03-26
8230202	Sonam Wangmo	Cleaning	2025-03-26
8230207	Tandin Wangmo	Cleaning	2025-03-26
8230222	Tshering Dollar	Lesson basic eletrical materials	2025-03-26
8230232	Ugyen Tenzin	Competition	2025-03-26
8240235	Yeshey Lhaden	Debate	2025-03-26
8240237	Yeshi Tshogay	Weeding	2025-03-26
8240249	Kelden Yoesel	Waste Collection	2025-03-26

11. Listing students and their clubs

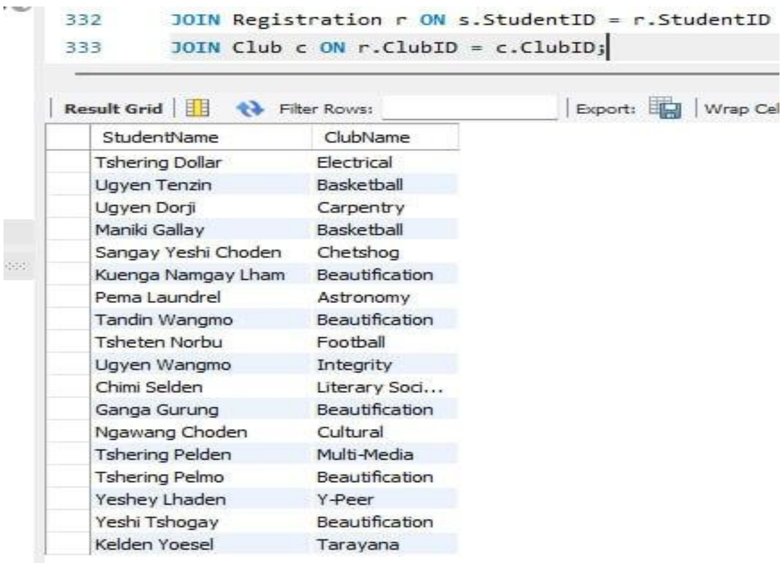


```sql
-- List students and their clubs

SELECT s.StudentName, c.ClubName
FROM Student s
JOIN Registration r ON s.StudentID = r.StudentID
JOIN Club c ON r.ClubID = c.ClubID;
```



```
JOIN Registration r ON s.StudentID = r.StudentID
JOIN Club c ON r.ClubID = c.ClubID;
```

StudentName	ClubName
Dorji Seldron	Beautification
Dorji Wangmo	Chetshog
Kuenzang Choden	Chetshog
Pema Rinchen	Cultural
Pema Yangchen	Democracy
Rinchen Dema	Beautification
Rinzin Wangmo	Beautification
Sonam Choden	Democracy
Sonam Wangmo	Beautification
Sonam Yoezer Densup	Multi-Media
Tandin Wangmo	Beautification
Tashi Lhamo	Chetshog
Tshering Dollar	Electrical
Ugyen Tenzin	Basketball
Ugyen Dorji	Carpentry
Maniki Gallay	Basketball
Sangay Yeshi Choden	Chetshog
Kuenga Namgay Lham	Beautification
Pema Laundrel	Astronomy



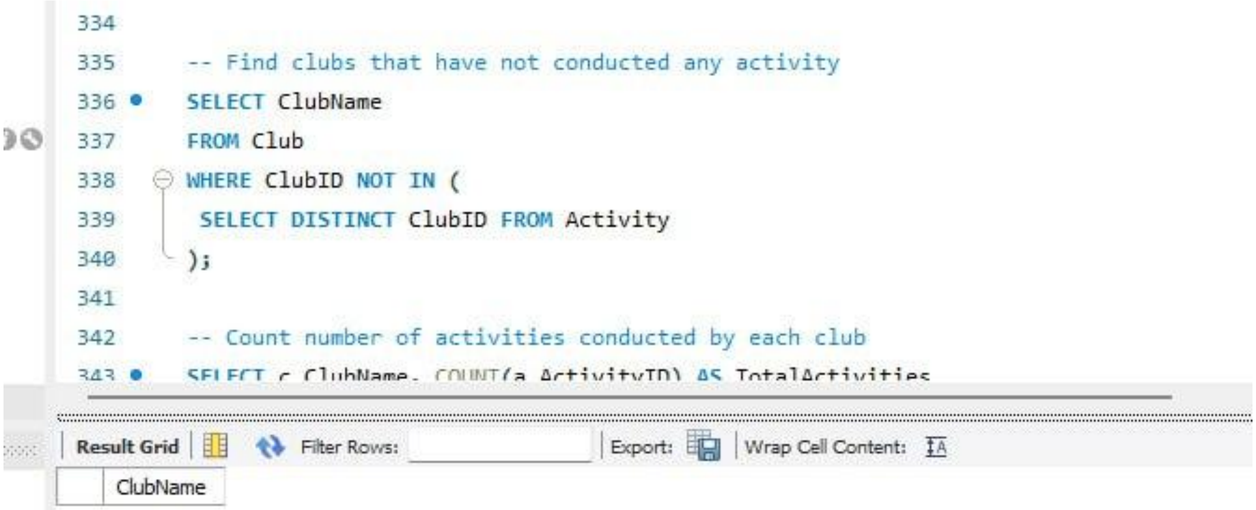
```
JOIN Registration r ON s.StudentID = r.StudentID
JOIN Club c ON r.ClubID = c.ClubID;
```

StudentName	ClubName
Tshering Dollar	Electrical
Ugyen Tenzin	Basketball
Ugyen Dorji	Carpentry
Maniki Gallay	Basketball
Sangay Yeshi Choden	Chetshog
Kuenga Namgay Lham	Beautification
Pema Laundrel	Astronomy
Tandin Wangmo	Beautification
Tsheten Norbu	Football
Ugyen Wangmo	Integrity
Chimi Selden	Literary Soci...
Ganga Gurung	Beautification
Ngawang Choden	Cultural
Tshering Pelden	Multi-Media
Tshering Pelmo	Beautification
Yeshey Lhaden	Y-Peer
Yeshi Tshogay	Beautification
Kelden Yoesel	Tarayana

Explanation

This SQL query is used to list all students along with the clubs they are registered in. It starts by selecting data from the Student table (aliased as s), which contains information about each student. The query then joins the Registration table (aliased as r), which acts as a link between students and clubs, using the StudentID field to match each student to their registration records. Next, it joins the Club table (aliased as c) on the ClubID field to retrieve the name of the club associated with each registration. The SELECT clause specifies that only the student's name and the club's name should be displayed. As a result, the output shows a list of students and the clubs they belong to, which is helpful for understanding student involvement in extracurricular activities.

12. Finding clubs that have not conducted any activity

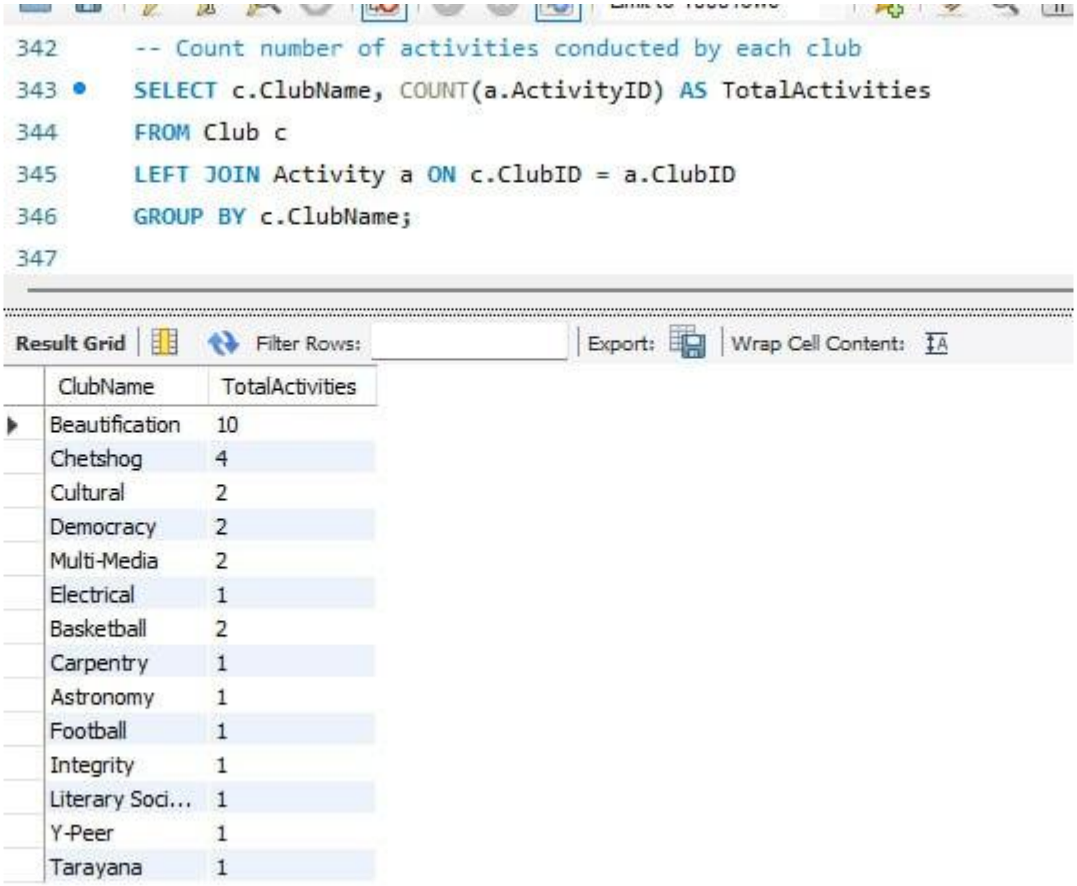


```

-- Find clubs that have not conducted any activity
SELECT ClubName
FROM Club
WHERE ClubID NOT IN (
 SELECT DISTINCT ClubID FROM Activity
);

-- Count number of activities conducted by each club
SELECT c.ClubName, COUNT(a.ActivityID) AS TotalActivities
```

Explanation:

The SQL query retrieves the names of clubs that have not conducted any activities. It does this by selecting ClubName from the Club table, but only for those clubs whose ClubID is not found in the list of ClubIDs from the Activity table. The subquery SELECT DISTINCT ClubID FROM Activity collects all the unique club IDs that have at least one activity recorded. The main query then filters out these clubs using the NOT IN condition, ensuring only the clubs without any associated activity are selected. This helps in identifying inactive clubs.

13. Counting number of activities conducted by each club



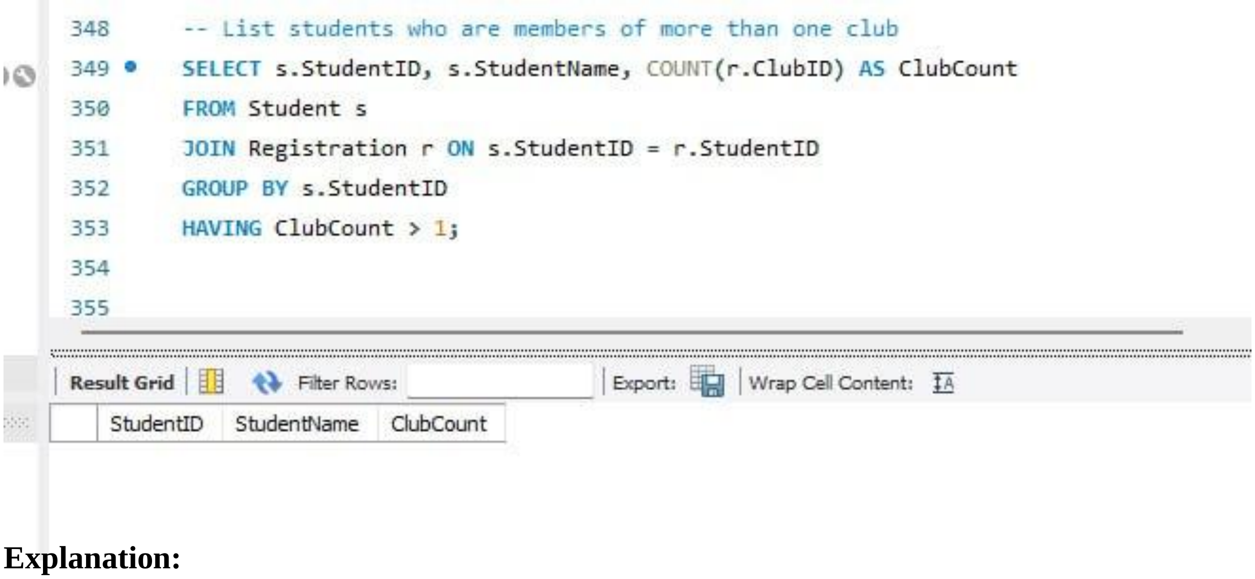
```sql
-- Count number of activities conducted by each club
SELECT c.ClubName, COUNT(a.ActivityID) AS TotalActivities
FROM Club c
LEFT JOIN Activity a ON c.ClubID = a.ClubID
GROUP BY c.ClubName;
```

ClubName	TotalActivities
Beautification	10
Chetshog	4
Cultural	2
Democracy	2
Multi-Media	2
Electrical	1
Basketball	2
Carpentry	1
Astronomy	1
Football	1
Integrity	1
Literary Soci...	1
Y-Peer	1
Tarayana	1

Explanation

This SQL query counts the number of activities conducted by each club and displays the club name along with the total number of activities. It uses a LEFT JOIN to combine the Club table (aliased as c) with the Activity table (aliased as a) based on the ClubID. The LEFT JOIN ensures that all clubs are included in the result, even if they haven't conducted any activities. The query then groups the results by ClubName and uses the COUNT function to count the number of ActivityIDs associated with each club. As a result, clubs with activities will show their total count, while clubs with no activities will show a count of zero.

14. Listing Students who are members of more than one club



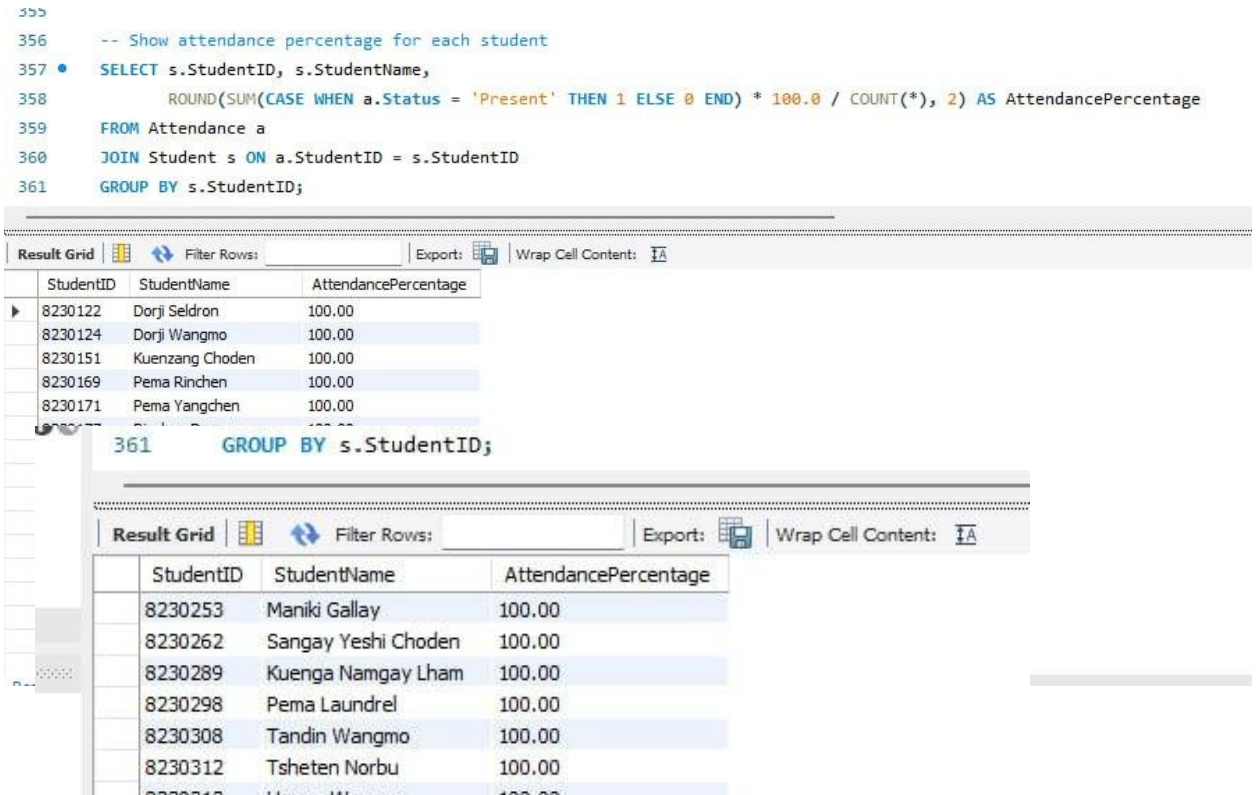
```
-- List students who are members of more than one club
SELECT s.StudentID, s.StudentName, COUNT(r.ClubID) AS ClubCount
FROM Student s
JOIN Registration r ON s.StudentID = r.StudentID
GROUP BY s.StudentID
HAVING ClubCount > 1;
```

StudentID	StudentName	ClubCount

Explanation:

This SQL query lists the students who are members of more than one club. It selects the student ID and name from the Student table and counts the number of clubs each student is registered in using the COUNT(r.ClubID) function. The JOIN operation connects the Student table with the Registration table based on matching StudentIDs. After joining, the query groups the records by StudentID to count how many clubs each student belongs to. Finally, the HAVING clause filters the grouped results to include only those students who are registered in more than one club (i.e., ClubCount > 1).

15. Showing attendance percentage for each student



```
-- Show attendance percentage for each student
SELECT s.StudentID, s.StudentName,
       ROUND(SUM(CASE WHEN a.Status = 'Present' THEN 1 ELSE 0 END) * 100.0 / COUNT(*), 2) AS AttendancePercentage
FROM Attendance a
JOIN Student s ON a.StudentID = s.StudentID
GROUP BY s.StudentID;
```

StudentID	StudentName	AttendancePercentage
8230122	Dorji Seldron	100.00
8230124	Dorji Wangmo	100.00
8230151	Kuenzang Choden	100.00
8230169	Pema Rinchen	100.00
8230171	Pema Yangchen	100.00

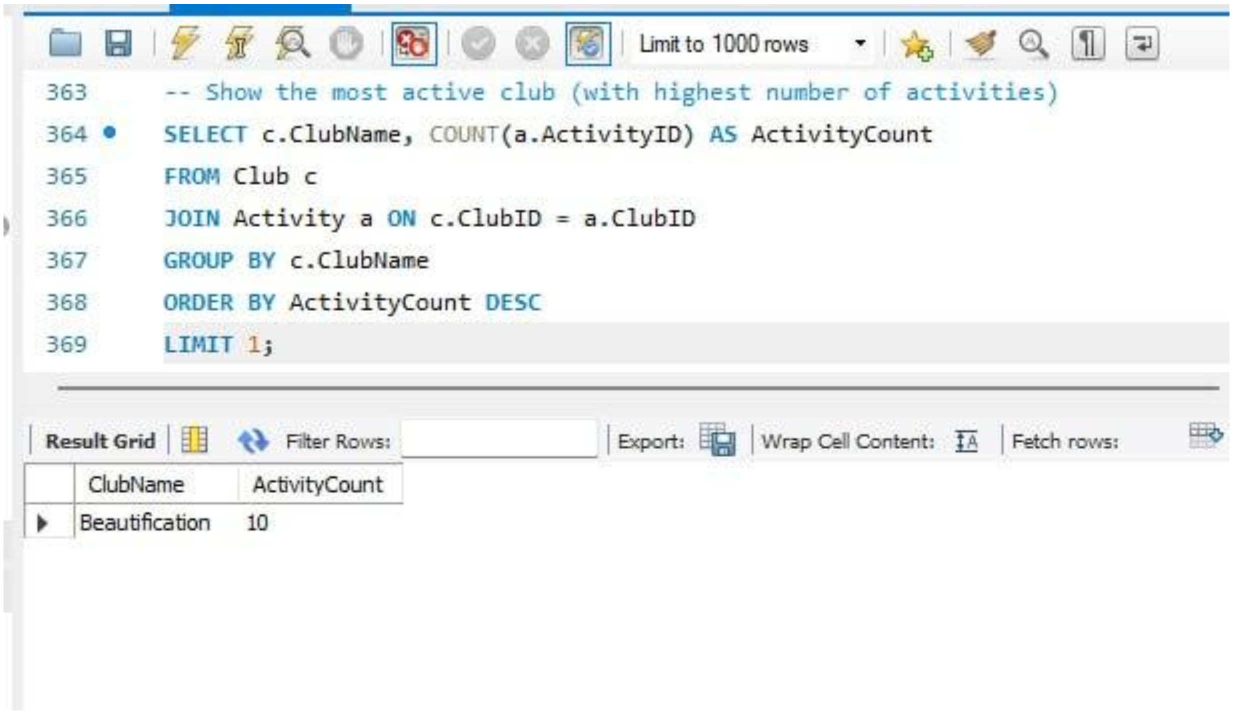
GROUP BY s.StudentID;

StudentID	StudentName	AttendancePercentage
8230253	Maniki Gallay	100.00
8230262	Sangay Yeshi Choden	100.00
8230289	Kuenga Namgay Lham	100.00
8230298	Pema Laundrel	100.00
8230308	Tandin Wangmo	100.00
8230312	Tsheten Norbu	100.00

Explanation:

This SQL query calculates and displays the attendance percentage for each student. It retrieves the student ID and name from the Student table and joins it with the Attendance table using the StudentID. Within the SELECT clause, it uses a CASE statement to count the number of times a student was marked as 'Present'. This value is then multiplied by 100.0 and divided by the total number of attendance records for that student (calculated using COUNT(*)). The ROUND function is used to round the result to two decimal places. Finally, the query groups the data by StudentID to calculate the percentage for each student individually.

16. Showing the most active Club



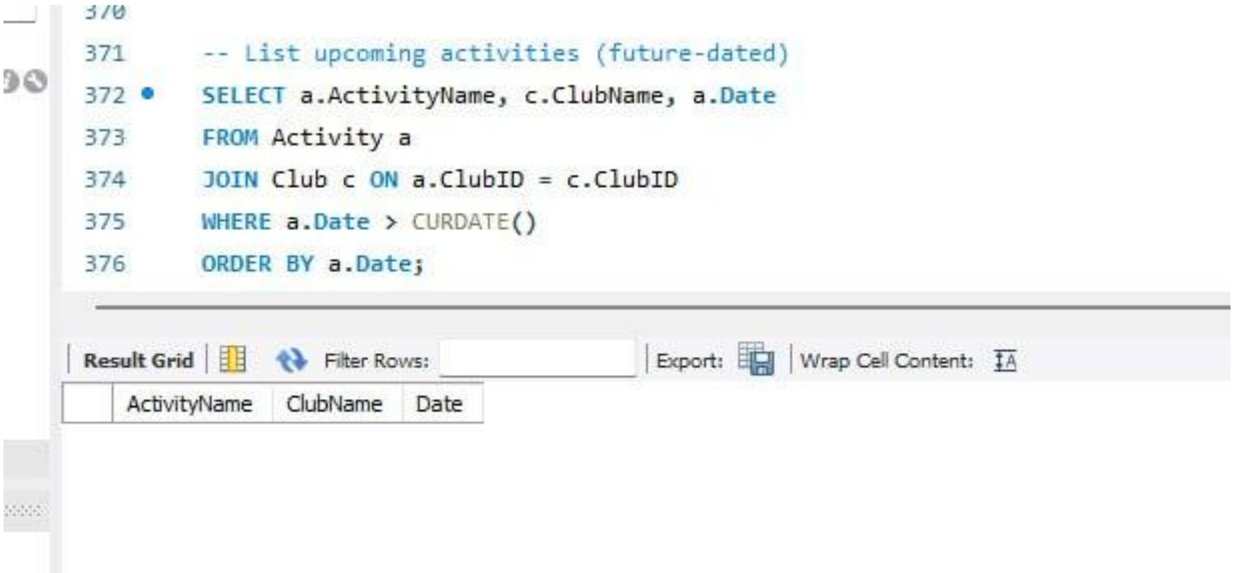
```
-- Show the most active club (with highest number of activities)
SELECT c.ClubName, COUNT(a.ActivityID) AS ActivityCount
FROM Club c
JOIN Activity a ON c.ClubID = a.ClubID
GROUP BY c.ClubName
ORDER BY ActivityCount DESC
LIMIT 1;
```

ClubName	ActivityCount
Beautification	10

Explanation

This SQL query identifies the most active club—the one that has conducted the highest number of activities. It joins the Club table with the Activity table using the ClubID to match each activity with its respective club. Then, it groups the data by ClubName and counts the number of ActivityIDs associated with each club to get the total activity count. The ORDER BY ActivityCount DESC sorts the results in descending order based on the number of activities, and LIMIT 1 ensures that only the club with the highest activity count is displayed.

17. Listing upcoming activities



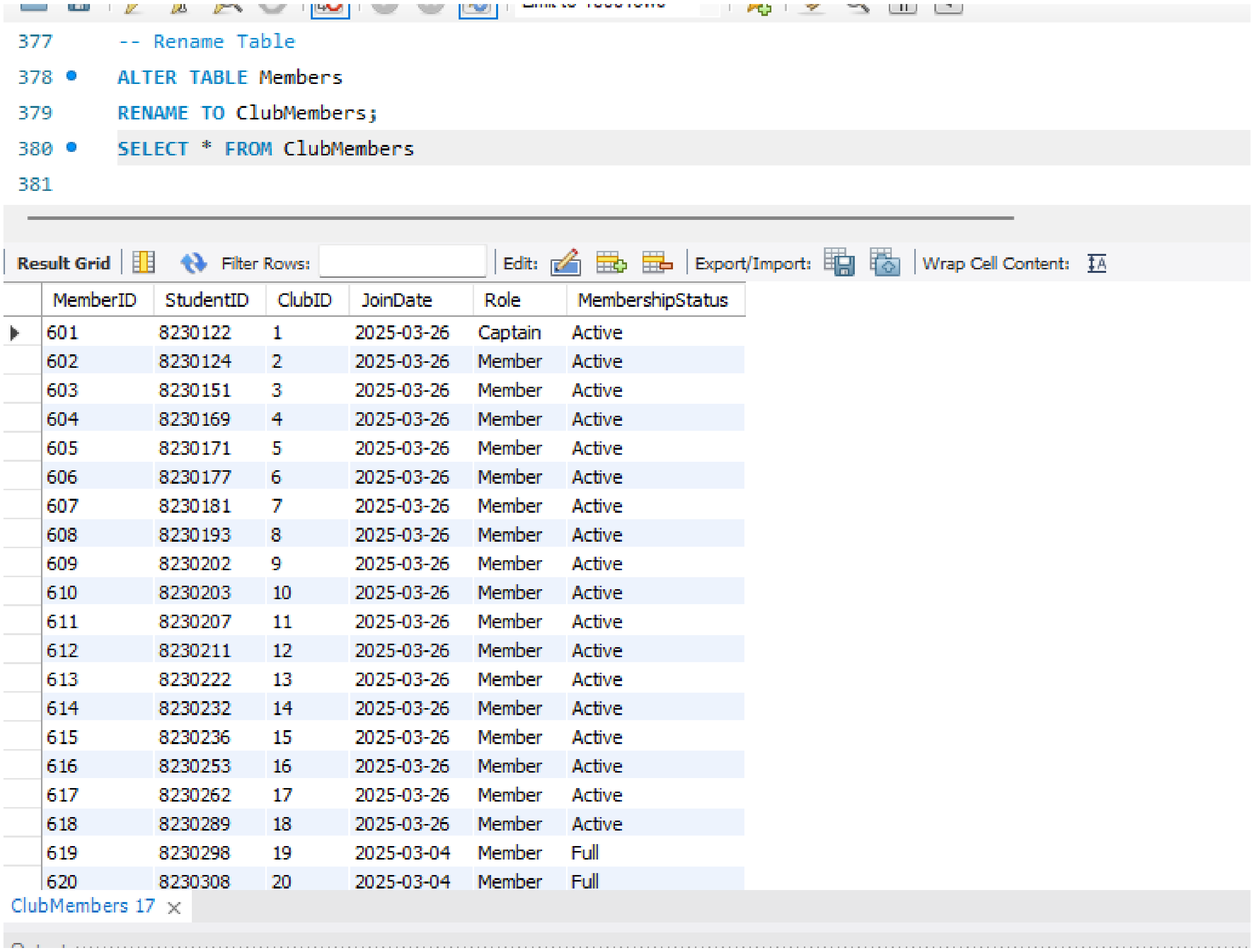
```sql
-- List upcoming activities (future-dated)
SELECT a.ActivityName, c.ClubName, a.Date
FROM Activity a
JOIN Club c ON a.ClubID = c.ClubID
WHERE a.Date > CURDATE()
ORDER BY a.Date;
```

	ActivityName	ClubName	Date

Explanation

This SQL query retrieves a list of upcoming activities by selecting the activity name, the club name, and the date of each activity. It does this by joining two tables: the Activity table, which contains details about each event, and the Club table, which holds information about the clubs hosting these activities. The join is based on the club's unique identifier to ensure each activity is matched with its respective club. To focus only on future events, the query filters out any activities that have already happened or are happening today by comparing the activity date to the current date. Finally, the results are ordered chronologically by the activity date, so the soonest upcoming activities appear first.

18. Renaming the Table

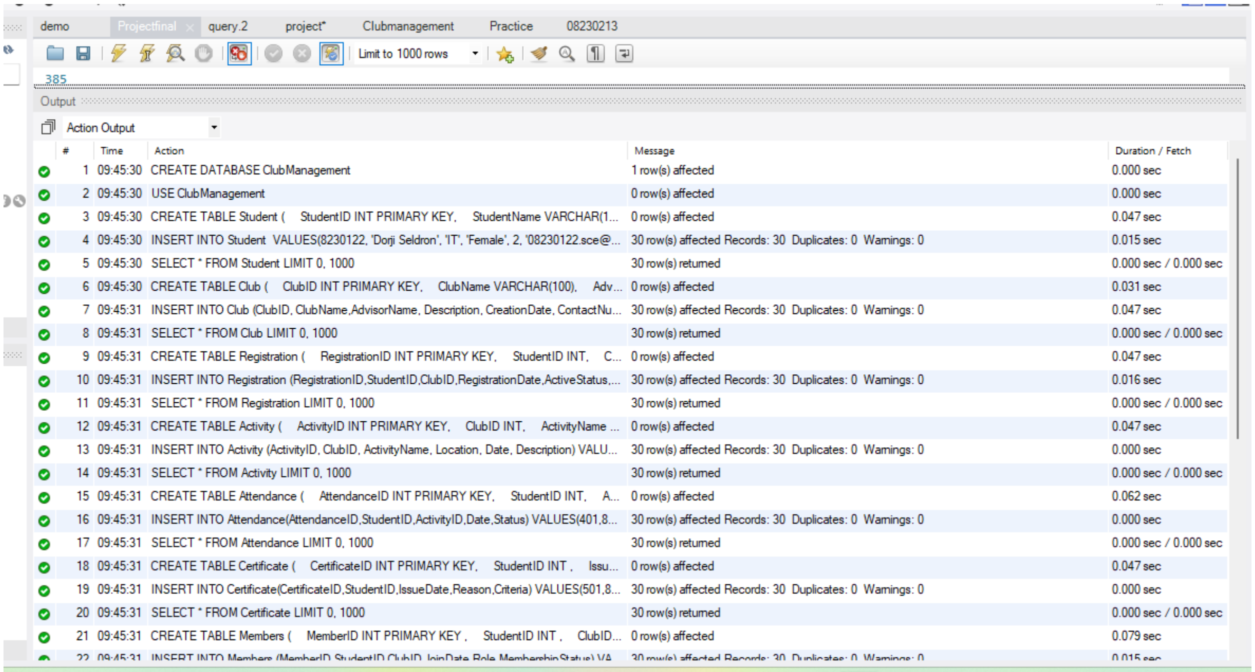


```sql
-- Rename Table
ALTER TABLE Members
RENAME TO ClubMembers;
SELECT * FROM ClubMembers
```

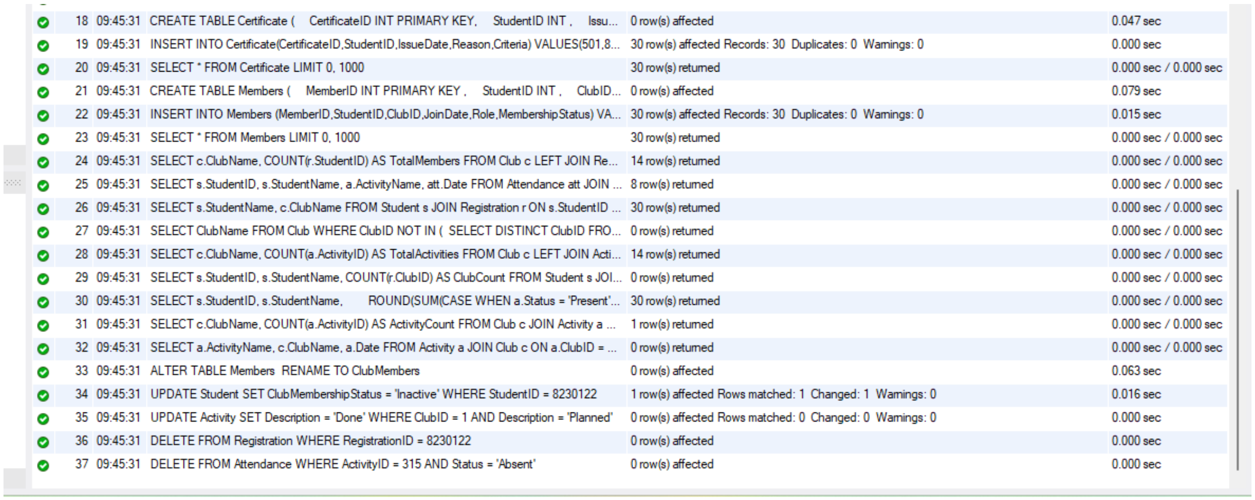
MemberID	StudentID	ClubID	JoinDate	Role	MembershipStatus
601	8230122	1	2025-03-26	Captain	Active
602	8230124	2	2025-03-26	Member	Active
603	8230151	3	2025-03-26	Member	Active
604	8230169	4	2025-03-26	Member	Active
605	8230171	5	2025-03-26	Member	Active
606	8230177	6	2025-03-26	Member	Active
607	8230181	7	2025-03-26	Member	Active
608	8230193	8	2025-03-26	Member	Active
609	8230202	9	2025-03-26	Member	Active
610	8230203	10	2025-03-26	Member	Active
611	8230207	11	2025-03-26	Member	Active
612	8230211	12	2025-03-26	Member	Active
613	8230222	13	2025-03-26	Member	Active
614	8230232	14	2025-03-26	Member	Active
615	8230236	15	2025-03-26	Member	Active
616	8230253	16	2025-03-26	Member	Active
617	8230262	17	2025-03-26	Member	Active
618	8230289	18	2025-03-26	Member	Active
619	8230298	19	2025-03-04	Member	Full
620	8230308	20	2025-03-04	Member	Full

Explanation:

This SQL query that renames the "Members" table to "ClubMembers," followed by a SELECT * query to retrieve all data from the newly named table. The result grid below successfully exhibits the ClubMembers table's data, including MemberID, StudentID, ClubID, JoinDate, Role (e.g., "Member," "Captain"), and MembershipStatus (e.g., "Active," "Full"), providing a clear view of 17 individual.



	#	Time	Action	Message	Duration / Fetch
✓	1	09:45:30	CREATE DATABASE ClubManagement	1 row(s) affected	0.000 sec
✓	2	09:45:30	USE ClubManagement	0 row(s) affected	0.000 sec
✓	3	09:45:30	CREATE TABLE Student (StudentID INT PRIMARY KEY, StudentName VARCHAR(1...	0 row(s) affected	0.047 sec
✓	4	09:45:30	INSERT INTO Student VALUES(8230122, 'Dorji Seldron', 'IT', 'Female', 2, '08230122.sce@...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.015 sec
✓	5	09:45:30	SELECT * FROM Student LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
✓	6	09:45:30	CREATE TABLE Club (ClubID INT PRIMARY KEY, ClubName VARCHAR(100), Adv...	0 row(s) affected	0.031 sec
✓	7	09:45:31	INSERT INTO Club (ClubID, ClubName,AdvisorName, Description, CreationDate, ContactNu...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.047 sec
✓	8	09:45:31	SELECT * FROM Club LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
✓	9	09:45:31	CREATE TABLE Registration (RegistrationID INT PRIMARY KEY, StudentID INT, C...	0 row(s) affected	0.047 sec
✓	10	09:45:31	INSERT INTO Registration (RegistrationID,StudentID,ClubID,RegistrationDate,ActiveStatus,...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.016 sec
✓	11	09:45:31	SELECT * FROM Registration LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
✓	12	09:45:31	CREATE TABLE Activity (ActivityID INT PRIMARY KEY, ClubID INT, ActivityName ...	0 row(s) affected	0.047 sec
✓	13	09:45:31	INSERT INTO Activity (ActivityID, ClubID, ActivityName, Location, Date, Description) VALU...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
✓	14	09:45:31	SELECT * FROM Activity LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
✓	15	09:45:31	CREATE TABLE Attendance (AttendanceID INT PRIMARY KEY, StudentID INT, A...	0 row(s) affected	0.062 sec
✓	16	09:45:31	INSERT INTO Attendance(AttendanceID,StudentID,ActivityID,Date,Status) VALUES(401,8...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
✓	17	09:45:31	SELECT * FROM Attendance LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
✓	18	09:45:31	CREATE TABLE Certificate (CertificateID INT PRIMARY KEY, StudentID INT , Issu...	0 row(s) affected	0.047 sec
✓	19	09:45:31	INSERT INTO Certificate(CertificateID,StudentID,IssueDate,Reason,Criteria) VALUES(501,8...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
✓	20	09:45:31	SELECT * FROM Certificate LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
✓	21	09:45:31	CREATE TABLE Members (MemberID INT PRIMARY KEY , StudentID INT , ClubID...	0 row(s) affected	0.079 sec
✓	22	09:45:31	INSERT INTO Members (MemberID,StudentID,ClubID,JoinDate,Role,MembershipStatus) VA...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.015 sec



	#	Time	Action	Message	Duration / Fetch
✓	18	09:45:31	CREATE TABLE Certificate (CertificateID INT PRIMARY KEY, StudentID INT , Issu...	0 row(s) affected	0.047 sec
✓	19	09:45:31	INSERT INTO Certificate(CertificateID,StudentID,IssueDate,Reason,Criteria) VALUES(501,8...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.000 sec
✓	20	09:45:31	SELECT * FROM Certificate LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
✓	21	09:45:31	CREATE TABLE Members (MemberID INT PRIMARY KEY , StudentID INT , ClubID...	0 row(s) affected	0.079 sec
✓	22	09:45:31	INSERT INTO Members (MemberID,StudentID,ClubID,JoinDate,Role,MembershipStatus) VA...	30 row(s) affected Records: 30 Duplicates: 0 Warnings: 0	0.015 sec
✓	23	09:45:31	SELECT * FROM Members LIMIT 0, 1000	30 row(s) returned	0.000 sec / 0.000 sec
✓	24	09:45:31	SELECT c.ClubName, COUNT(r.StudentID) AS TotalMembers FROM Club c LEFT JOIN Re...	14 row(s) returned	0.000 sec / 0.000 sec
✓	25	09:45:31	SELECT s.StudentID, s.StudentName, a.ActivityName, att.Date FROM Attendance att JOIN ...	8 row(s) returned	0.000 sec / 0.000 sec
✓	26	09:45:31	SELECT s.StudentName, c.ClubName FROM Student s JOIN Registration r ON s.StudentID ...	30 row(s) returned	0.000 sec / 0.000 sec
✓	27	09:45:31	SELECT ClubName FROM Club WHERE ClubID NOT IN (SELECT DISTINCT ClubID FRO...	0 row(s) returned	0.000 sec / 0.000 sec
✓	28	09:45:31	SELECT c.ClubName, COUNT(a.ActivityID) AS TotalActivities FROM Club c LEFT JOIN Acti...	14 row(s) returned	0.000 sec / 0.000 sec
✓	29	09:45:31	SELECT s.StudentID, s.StudentName, COUNT(r.ClubID) AS ClubCount FROM Student s JOI...	0 row(s) returned	0.000 sec / 0.000 sec
✓	30	09:45:31	SELECT s.StudentID, s.StudentName, ROUND(SUM(CASE WHEN a.Status = 'Present'...	30 row(s) returned	0.000 sec / 0.000 sec
✓	31	09:45:31	SELECT c.ClubName, COUNT(a.ActivityID) AS ActivityCount FROM Club c JOIN Activity a ...	1 row(s) returned	0.000 sec / 0.000 sec
✓	32	09:45:31	SELECT a.ActivityName, c.ClubName, a.Date FROM Activity a JOIN Club c ON a.ClubID = ...	0 row(s) returned	0.000 sec / 0.000 sec
✓	33	09:45:31	ALTER TABLE Members RENAME TO ClubMembers	0 row(s) affected	0.063 sec
✓	34	09:45:31	UPDATE Student SET ClubMembershipStatus = 'Inactive' WHERE StudentID = 8230122	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec
✓	35	09:45:31	UPDATE Activity SET Description = 'Done' WHERE ClubID = 1 AND Description = 'Planned'	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.000 sec
✓	36	09:45:31	DELETE FROM Registration WHERE RegistrationID = 8230122	0 row(s) affected	0.000 sec
✓	37	09:45:31	DELETE FROM Attendance WHERE ActivityID = 315 AND Status = 'Absent'	0 row(s) affected	0.000 sec

9. Implementation of Tutor Feedback

Throughout the development of our project, we received valuable feedback from our tutor, which greatly contributed to improving both the technical accuracy and presentation quality of our work. One major area of feedback was the object diagram, where the tutor pointed out that our initial version lacked precision and did not clearly represent the actual instances of the classes during runtime. In response, we revised the diagram to reflect the correct object relationships and ensured consistency with our class diagram.

Additionally, the tutor advised us to make all UML diagrams more visible and readable. We addressed this by increasing font sizes, adding proper labels, using standard UML notations, and ensuring that each diagram (class, object, use case, and sequence) was clearly formatted and easy to interpret. This enhanced the clarity of our documentation, especially when printed or projected.

The tutor also emphasized the need to improve consistency between diagrams. For example, certain attributes and methods shown in the class diagram were missing or inconsistent in the object diagram. We cross-checked all diagrams and updated them to maintain alignment across all models.

Moreover, we were encouraged to justify our design choices. With the tutor’s guidance, we added short descriptions below each diagram to explain its purpose and the logic behind key decisions such as relationships, multiplicity, and role of each class.

The tutor also helped us understand how to apply realistic naming conventions and improve diagram layout for better flow and readability. Based on this, we renamed some class and object elements and rearranged diagram elements to follow a logical left-to-right or top-down sequence.

Overall, our tutor's feedback played a crucial role in refining our design and improving the overall quality of our project documentation. It helped us not only fix specific issues but also understand best practices for professional modeling and communication in software development.

10. Conclusion

The Club Management System project marks a significant achievement in applying theoretical knowledge to a practical, real-world scenario. The primary aim of this system was to streamline and automate the process of managing student club registrations, maintaining membership records, tracking participation, and managing club-related data efficiently. By designing and implementing a well-structured relational database, supported by clear UML diagrams, we were able to develop a system that reduces redundancy, increases data accuracy, and enhances usability for both administrators and students.

The project involved creating key tables such as Student, Club, and Registration, along with implementing data integrity through the use of primary keys, foreign key constraints, and checks. We also applied normalization techniques up to the third normal form (3NF) to eliminate anomalies and ensure the database structure was logically sound and efficient.

In addition to the technical implementation, this project provided us with valuable learning experiences in terms of team collaboration, time management, and iterative development. Feedback from our tutor played a crucial role in refining our object diagram, improving the visibility and clarity of our UML diagrams, and aligning our system design with best practices. This process helped us better understand the importance of feedback, critical thinking, and continuous improvement in software development.

Moreover, this system has the potential for future enhancement, such as integrating a web-based user interface, login functionality for club coordinators, attendance tracking, and automated email notifications. The scalability and modularity of the current design make it feasible to expand the system based on institutional needs.

Overall, the Club Management System not only met the core objectives but also served as a platform for us to apply database concepts, strengthen our problem-solving skills, and gain insight into real-world software development workflows. It stands as a useful solution for educational institutions seeking to promote and manage student involvement in extracurricular activities more effectively.