Education Industry

Name: Hann Zhao
Student ID: 917565433
Github username: hazhao33

Version History Table:

Checkpoint #	Date Submitted
Checkpoint 1	6/16/23
Checkpoint 2	6/28/23
Checkpoint 3	7/13/23

Table of Content

Project Description

The School Database Management System is designed to efficiently organize and manage staff and student information within a school. This system aims to streamline administrative tasks and ensures accurate record keeping of both staff and students. Ultimately improving the overall efficiency and effectiveness of the school's operations without having to create their own database management system. This school database management system aims to provide an efficient solution to all educational institutions to help with staff management, student management, attendance tracking, class scheduling, and access management. The staff management creates a centralized database of employee information such as personal details, contact information, qualifications, employment history, and attendance records. The student management allows administrators to keep track of student information, enrollment, and academic records. Attendance tracking keeps track of staff and students' daily attendance, leaves, and absences. Class scheduling allows schools to create and optimize class scheduling with minimizing conflicts and improving scheduling accuracy. Access management authorizes role based access control to the whole school database management, which ensures sensitive data is only accessible only to authorized staff. The school database management system provides a solution for educational institutions to manage staff and student information and allows them to focus on providing quality education. The two software that is able to use this database is Ilearn and Canvas, which are both educational software. With this school database management system, those two software can provide all in one solutions to schools from school staffing to having online classes.

Functional Database Requirements

1. Institutions
   - 1.1. An institution shall have many registered staff.
   - 1.2. An institution shall have many registered students.
   - 1.3. An institution shall have many registered administrators.
   - 1.4. An institution shall have many roles.
   - 1.5. An institution shall have many departments.
   - 1.6. An institution shall have many classes.
   - 1.7. An institution shall have many majors.
   - 1.8. An institution shall have many clubs.
   - 1.9. An institution shall have many sports teams.
   - 1.10. An institution shall have many semesters.
   - 1.11. An institution shall have many events.
   - 1.12. An institution shall have many events.
   - 1.13. An institution shall have many books.
2. User
   - 2.1. A user shall create only one account.
   - 2.2. A user shall be added by at most one administrator.
   - 2.3. A user shall be removed by at most one administrator.
3. Administrator
   - 3.1. An administrator is a user.
   - 3.2. An administrator shall belong to at least one institution.
   - 3.3. An administrator shall have the capability to change many user departments.
   - 3.4. An administrator shall be able to create multiple new departments.
   - 3.5. An administrator shall be able to add many administrators.
4. Staff
   - 4.1. A staff shall belong to at least one institution.
   - 4.2. A staff is a user.
   - 4.3. Staff shall be assigned to many departments.
   - 4.4. Staff shall be assigned to many classes.
   - 4.5. Staff shall have many qualifications.
   - 4.6. Staff shall have many employment histories.
   - 4.7. Staff shall be assigned to many majors.
5. Student
   - 5.1. A student is a user.
   - 5.2. A student shall belong to at least one institution.
   - 5.3. A student shall register for many classes.
   - 5.4. A student shall register for many majors.
   - 5.5. A student shall register for many semesters.
   - 5.6. A student shall have only one academic record.
   - 5.7. A student shall have only one medical history.
   - 5.8. A student shall select one default payment.

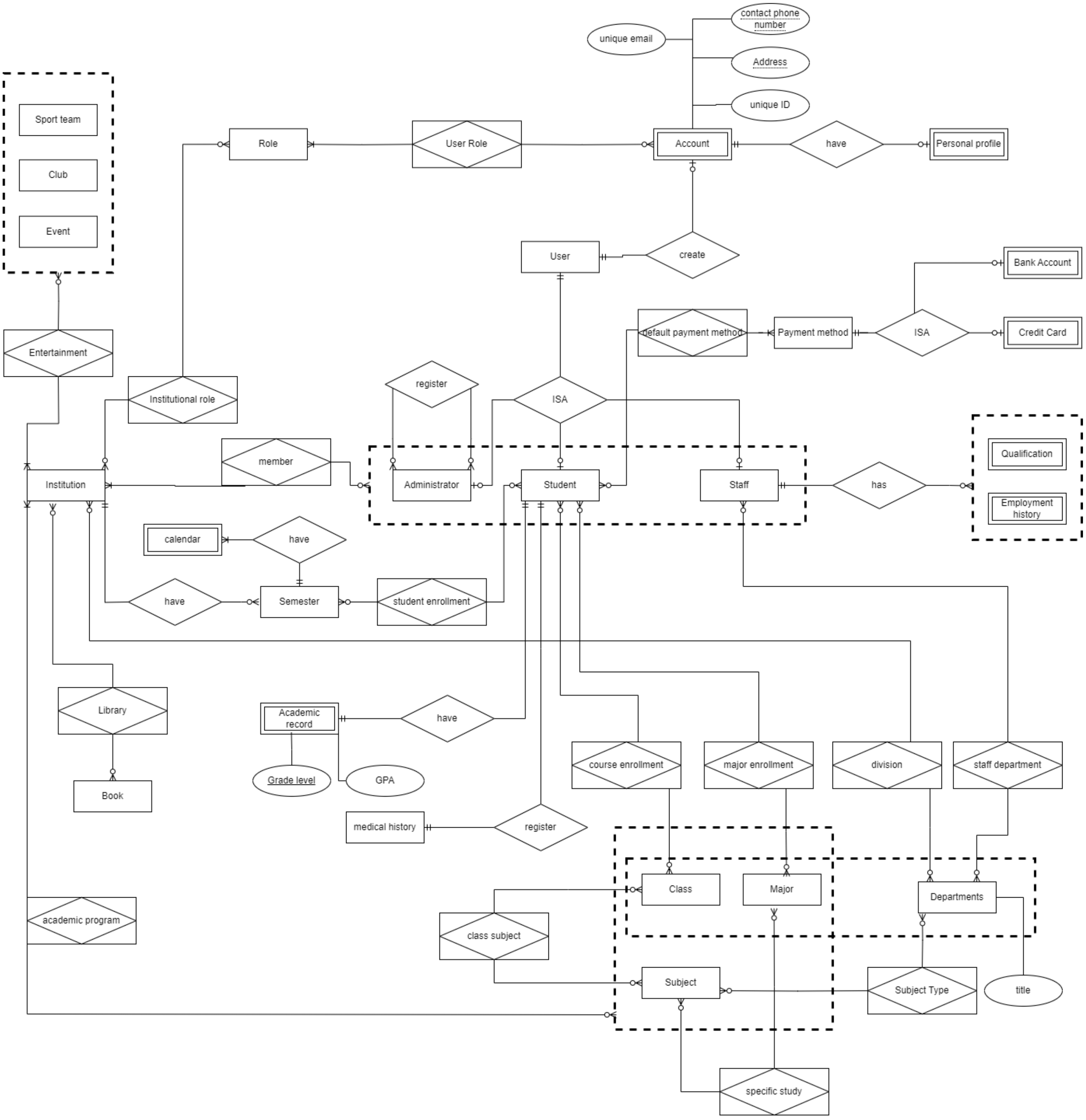
6. Major
    6.1. A major shall be assigned to many students.
    6.2. A major shall be assigned to many staff.
    6.3. A major shall have many subjects.
    6.4. A major shall belong to at least one institution.
7. Role
    7.1. A role shall be assigned to many accounts.
    7.2. A role shall be created by at least one administrator.
8. Class
    8.1. A class shall be assigned to many staff members.
    8.2. A class shall have many students.
    8.3. A class shall be assigned to many subject.
    8.4. A class shall belong to at least one institution.
9. Subject
    9.1. A subject shall be assigned to many department.
    9.2. A subject shall have many classes.
    9.3. A subject shall belong to many majors.
10. Department
    10.1. The department shall belong to many institutions.
    10.2. A department shall have only one title.
    10.3. A department shall have many staff.
    10.4. A department shall have many subjects.
11. Account
    11.1. An account shall belong to only one user.
    11.2. An account shall have only one unique ID assigned.
    11.3. An account shall be assigned to only one unique email.
    11.4. An account shall have only one name.
    11.5. An account shall have at least one address.
    11.6. An account shall be assigned to at most one role.
    11.7. An account shall have at least one contact phone number.
    11.8. An account shall be able to edit at most one personal profile.
12. Personal profile
    12.1. A profile shall be owned by only one account
13. Role
    13.1. A role shell has many accounts.
    13.2. A role shell belongs to many institutions.
14. Medical history
    14.1. A medical history shall be owned by only one student.
15. Academic record
    15.1. An academic record shall have one GPA.
    15.2. An academic record shall have one grade level.
16. Qualifications
    16.1. Qualification shall be owned by only one staff.

17. employment history
    17.1. An employment history shall be owned by only one staff.
18. Club
    18.1. A club shall belong to at least one institution.
19. Sport team
    19.1. A sports team shall belong to at least one institution.
20. Payment Method
    20.1. A payment method is either a bank account or a credit card.
21. Semester
    21.1. A semester shall have many students.
    21.2. A semester shall belongs to many institutions.
    21.3. A semester shall have at least one calendar.
22. Event
    22.1. A Event shall belong to at least one institution.
23. Book
    23.1. A book shall be found in many institution.
24. Calendar
    24.1. A calendar shall belong to a semester.

Non-functional Database Requirements

1. Performance
    1.1. The database shall be responsive.
    1.2. All changes and tools shall be performed in a timely manner.
    1.3. The database shall have minimal downtime.
    1.4. The database shall be able to handle the increasing number of users.
    1.5. The database shall handle a large number of requests at once.
    1.6. The database shall be able to process large amounts of data at once.
2. Usability
    2.1. The database interface shall be user friendly.
3. Security
    3.1. Only storing hashed passwords on the database system.
    3.2. The database shall censure sensitive data from all users is securely stored.
    3.3. The database shall have a robust access control system.
    3.4. The system shall have encryption support during transmission.
4. Scalability
    4.1. The database shall handle an increasing number of members.
    4.2. The database shall support the growth of data.
5. Backup
    5.1. The database shall be backed up daily.
    5.2. The database shall support data recovery.
6. Accessibility
    6.1. The database shall be compatible with many operating systems.
    6.2. The system shall be compatible with all popular web browsers.
7. System Requirements
    7.1. The system should be able to integrate with another existing system.
8. Storage
    8.1. The database shall have sufficient storage capacity for future growth.
    8.2. The database shall support data archiving.
9. Environmental
    9.1. The system shall be designed to minimize power usage.
10. Legal
    10.1. The system shall have data privacy protection for all users.
11. Organizational
    11.1. The system shall have an efficient organization of data.
    11.2. The system shall support the customization of the system for each institution.

Entity Relationship Diagram (ERD)



Entity Description

1. Role (Strong)
   * role_id: key, numeric
   * role_name: composite, alphanumeric
   * description: alphanumeric
2. User (Strong)
   * user_id: key, numeric
   *administrator_id: key, numeric
   * account_id: key, numeric
   *student_id: key, numeric
   *staff_id: key, numeric
3. Sports Team (Strong)
   * team_id: key, numeric
   *team_name: composite, alphanumeric
   *description: alphanumeric
4. Club (Strong)
   * club_id: key, numeric
   *club_name: composite, alphanumeric
   *description: alphanumeric
5. Payment Method (Strong)
   * payment_method: key, numeric
   * bankAcc_id: key, numeric
   *creditCard_id: key, numeric
6. Institution (Strong)
   * institution_id: key, numeric
   *institution_name: composite, alphanumeric
   *description: alphanumeric
7. Administrator (Strong)
   * administrator_id: key, numeric
   * name: composite, alphanumeric
   * age: key, numeric
   * register_by: key, numeric
8. Student (Strong)
   * student_id: key, numeric
   * name: composite, alphanumeric
   * dob: composite, alphanumeric
9. Staff (Strong)
   * staff_id: key, numeric
   * name: composite, alphanumeric
   * dob: composite, alphanumeric
10. Semester(Strong)
    * semester_id: key, numeric

* year: multivalue, timestamp
* type: composite, alphanumeric
* institution: key, numeric

11. Class (Strong)
    * class_id: key, numeric
    * class_name: composite, alphanumeric
    * seat_limit: key, numeric
    * price: key, numeric
12. Major (Strong)
    * major_id: key, numeric
    * major_name: composite, alphanumeric
    * sudent_count: key, numeric
13. Departments (Strong)
    * department_id: key, numeric
    * title: composite, alphanumeric
    * staff_num: key, numeric
14. Event(Strong)
    * event_id: key, numeric
    * title: composite, alphanumeric
    * location: composite, alphanumeric
    * description: alphanumeric
15. Book (Strong)
    * book_id: key, numeric
    * ISBM: key, alphanumeric
    * name: composite, alphanumeric
    * author: composite, alphanumeric
16. Subject(Strong)
    * subject_id: key, numeric
    * title: composite, alphanumeric
    * description: alphanumeric
17. Medical History (Strong)
    * medical_history__id: key, numeric
    * time: composite, alphanumeric
    * description: alphanumeric
    * student_id: key, numeric
18. Profile (weak)
    * profile_id: key, numeric
    * dob: composite, alphanumeric
    * account_id: key, numeric
19. Library (weak)
    * library_id: key, numeric
    * lib_name: composite, alphanumeric
    * location: composite, alphanumeric

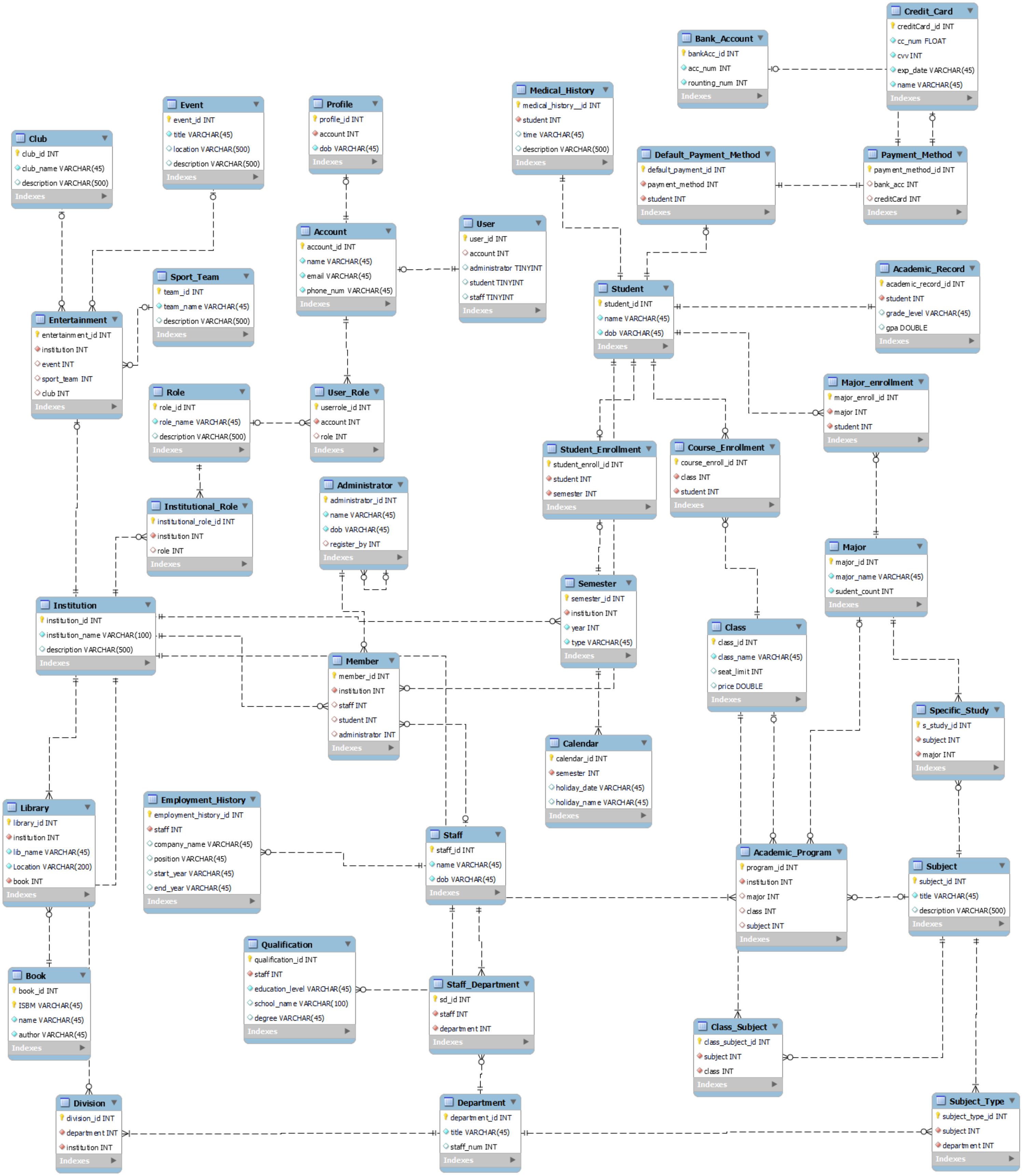
   * institution_id: key, numeric
   * book_id: key, numeric
20. User Role (weak)
   * userrole_id: key, numeric
   * role_id: key, numeric
   * account_id: key, numeric
21. Account (weak)
   * account_id: key, numeric
   * name: composite, alphanumeric
   * email: composite, alphanumeric
   * phone_num: multivalue, alphanumeric
22. Default payment method (weak)
   * default_payment_id: key, numeric
   * payment_method: key, numeric
   * student_id: key, numeric
23. Bank Account (Weak)
   * bankAcc_id: key, numeric
   * acc_num: key, numeric
   * rounting_num: key, numeric
24. Credit Card (Weak)
   * creditCard_id: key, numeric
   * cc_num: key, numeric
   * name: composite, alphanumeric
   * cvv: key, numeric
   * exp_date: multivalue, timestamp
25. Institutional Role (Weak)
   * institutional_role_id: key, numeric
   * institution_id: key, numeric
   * role_id: key, numeric
26. Qualification (Weak)
   * qualification_id: key, numeric
   *staff: key, numeric
   * education_level: composite, alphanumeric
   * school_name: composite, alphanumeric
   * degree: composite, alphanumeric
27. Employment History (Weak)
   * employment_history_id: key, numeric
   * company_name: composite, alphanumeric
   * start_year: composite, alphanumeric
   * end_year: composite, alphanumeric
   * position: composite, alphanumeric
   * staff: key, numeric
28. Student enrollment (Weak)

   * student_enroll_id: key, numeric
   * student_id: key, numeric
   * semester_id: key, numeric
29. Academic Record (Weak)
   * academic_record_id: key, numeric
   * grade_level: composite, alphanumeric
   * student_id: key, numeric
   *gpa: key, numeric
30. Course enrollment (Weak)
   * course_enroll_id: key, numeric
   * student_id: key, numeric
   * class_id: key, numeric
31. Major enrollment (Weak)
   * major_enroll_id: key, numeric
   * major_id: key, numeric
   * student_id: key, numeric
32. Division (Weak)
   * division_id: key, numeric
   * department_id: key, numeric
   * institution_id: key, numeric
33. Staff Department (Weak)
   * sd_id: key, numeric
   * staff_id: key, numeric
   * department_id: key, numeric
34. Academic Program (Weak)
   * program_id: key, numeric
   * major_id: key, numeric
   * class_id: key, numeric
   * institution_id: key, numeric
   * subject_id: key, numeric
35. Specific Study (Weak)
   * s_study_id: key, numeric
   *subject_id: key, numeric
   *major_id: key, numeric
36. Member (Weak)
   * member_id: key, numeric
   * institution_id: key, numeric
   * staff_id: key, numeric
   * student_id: key, numeric
   * administrator_id: key, numeric
37. Entertainment (Weak)
   * entertainment_id: key, numeric

* institution_id: key, numeric
*event_id: key, numeric
* team_id: key, numeric
* club_id: key, numeric

38. Calendar (Weak)
    * calendar_id: key, numeric
    * semester: key, numeric
    * holiday_date: composite, alphanumeric
    * holiday_name: composite, alphanumeric
39. Subject type (Weak)
    * subject_type_id: key, numeric
    * subject_id: key, numeric
    * major_id: key, numeric
40. Class Subject(Weak)
    * class_subject_id: key, numeric
    * subject_id: key, numeric
    * class_id: key, numeric

Entity Establishment Relationship Diagram (ERR)



Constraints Description

Table	FK	ON DELETE	ON UPDATE	COMMENT
User	account	ON CASCADE	ON CASCADE	If an account is deleted, then the user from that account must be deleted as well.
Payment_ Method	bank_ account	SET NULL	ON CASCADE	If a bank account is removed from the payment method, it will be set to null until a new bank account is assigned.
Payment_ Method	credit_ card	SET NULL	ON CASCADE	If a credit card is removed from the payment method, it will be set to null until a new credit card is assigned.
Administrator	administrator	ON CASCADE	NO ACTION	If the FK administrator is deleted, it will still remain in the administrator table record.
Semester	institution	ON CASCADE	ON CASCADE	If an institution is deleted, then the semester from that institution must be deleted as well.
Profile	account	ON CASCADE	ON CASCADE	If an account is deleted, then the profile from that account must be deleted as well.
Library	institution	ON CASCADE	ON CASCADE	If an institution is deleted, then the library from that institution must be deleted as well.
Library	book	ON CASCADE	ON CASCADE	If a book is deleted, then the library from that book must be deleted as well.
User_Role	role	SET NULL	ON CASCADE	If a role is removed from the user_role, it will be set to null until a new role is assigned.
User_Role	account	ON CASCADE	ON CASCADE	If an account is deleted, then the user-role from that account must be deleted as well.
Default_ Payment_ Method	student	ON CASCADE	ON CASCADE	If a student is deleted, then the default_payment_method from that student must be deleted as well.
Default_ Payment_ Method	payment_ method	ON CASCADE	ON CASCADE	If a payment_method is deleted, then the default_payment_method from that payment_method must be deleted as well.
Institution_ Role	institution	ON CASCADE	ON CASCADE	If an institution is deleted, then the institution_role from that institution must be deleted as well.
Institution_ Role	role	SET NULL	ON CASCADE	If a role is removed from the institution_role, it will be set to null until a new role is assigned.
Qualification	staff	ON CASCADE	ON CASCADE	If a staff is deleted, then the qualification from that staff must be deleted as well.

Employment_History	staff	ON CASCADE	ON CASCADE	If a staff is deleted, then the employment_history from that staff must be deleted as well.
Student_Enrollment	student	ON CASCADE	ON CASCADE	If a student is deleted, then the student_enrollment from that student must be deleted as well.
Student_Enrollment	semester	ON CASCADE	ON CASCADE	If a semester is deleted, then the student_enrollment from that semester must be deleted as well.
Academic_Record	student	ON CASCADE	ON CASCADE	If a student is deleted, then the academic_record from that student must be deleted as well.
Medical_History	student	ON CASCADE	ON CASCADE	If a student is deleted, then the medical_history from that student must be deleted as well.
Course_Enrollment	student	ON CASCADE	ON CASCADE	If a student is deleted, then the course_enrollment from that student must be deleted as well.
Course_Enrollment	class	ON CASCADE	ON CASCADE	If a class is deleted, then the course_enrollment from that class must be deleted as well.
Major_Enrollment	major	ON CASCADE	ON CASCADE	If a major is deleted, then the major_enrollment from that major must be deleted as well.
Major_Enrollment	student	ON CASCADE	ON CASCADE	If a student is deleted, then the major_enrollment from that student must be deleted as well.
Division	department	ON CASCADE	ON CASCADE	If a department is deleted, then the division from that department must be deleted as well.
Division	institution	ON CASCADE	ON CASCADE	If an institution is deleted, then the division from that institution must be deleted as well.
Staff_Department	staff	ON CASCADE	ON CASCADE	If a staff is deleted, then the staff_department from that staff must be deleted as well.
Staff_Department	department	ON CASCADE	ON CASCADE	If a department is deleted, then the staff_department from that department must be deleted as well.
Academic_Program	major	SET NULL	ON CASCADE	If a major is removed from the academic_program, it will be set to null until a new major is assigned.
Academic_Program	class	SET NULL	ON CASCADE	If a class is removed from the academic_program, it will be set to null until a new class is assigned.
Academic_Program	institution	ON CASCADE	ON CASCADE	If an institution is deleted, then the academic_program from that institution must be deleted as well.
Academic_Program	subject	SET NULL	ON CASCADE	If a subject is removed from the academic_program, it will be set to null until a new subject is assigned.
Specific_Study	subject	ON CASCADE	ON CASCADE	If a subject is deleted, then the specific_study from that subject must be deleted as well.
Specific_	major	ON	ON	If a major is deleted, then the specific_study from that major

Study		CASCADE	CASCADE	must be deleted as well.
Member	institution	ON CASCADE	ON CASCADE	If an institution is deleted, then the member from that institution must be deleted as well.
Member	staff	SET NULL	ON CASCADE	If a staff is removed from the member, it will be set to null until new staff is assigned.
Member	student	SET NULL	ON CASCADE	If a student is removed from the member, it will be set to null until a new student is assigned.
Member	administrator	SET NULL	ON CASCADE	If an administrator is removed from the member, it will be set to null until a new administrator is assigned.
Entertainment	institution	ON CASCADE	ON CASCADE	If an institution is deleted, then the entertainment from that institution must be deleted as well.
Entertainment	event	SET NULL	ON CASCADE	If an event is removed from the entertainment, it will be set to null until a new event is assigned.
Entertainment	sport_team	SET NULL	ON CASCADE	If a sport_team is removed from the entertainment, it will be set to null until a new sport_team is assigned.
Entertainment	club	SET NULL	ON CASCADE	If a club is removed from the entertainment, it will be set to null until a new club is assigned.
Calendar	semester	ON CASCADE	ON CASCADE	If a semester is deleted, then the calendar from that semester must be deleted as well.
Subject_Type	subject	ON CASCADE	ON CASCADE	If a subject is deleted, then the subject_type from that subject must be deleted as well.
Subject_Type	department	ON CASCADE	ON CASCADE	If a department is deleted, then the subject_type from that department must be deleted as well.
Class_Subject	subject	ON CASCADE	ON CASCADE	If a subject is deleted, then the class_subject from that subject must be deleted as well.
Class_Subject	class	ON CASCADE	ON CASCADE	If a class is deleted, then the class_subject from that class must be deleted as well.