# College Database Project

Pablo Martinez Latorre

Student ID: 922270801

GitHub Username: pabloSFSU


| Milestone/Version | Date |
|---|---|
| M1V1 | 14 Sept. 2021 |
| M1V2 | 25 Sept. 2021 |
| M1V3 | 30 Sept. 2021 |
| M1V4 | 4 Oct. 2021 |
| M2V1 | 1 Nov 2021 |

# Contents

**Section I:** Project Description……..... 3

**Section II:** Use Cases……. 4

**Section III:** Database Requirements…….. 7

**Section IV:** Main Entities, Attributes and Keys……..... 13

**Section V:** Entity Relationship Diagram…… 27

**Section VI:** Testing Table…….. 28

**Section VII:** Database Model / EER……. 32

**Section XI:** Testing Table……. 34

# **Section I:** Project Description

---

The aim of this project is to build a cutting-edge College Database Management System to enable universities an easier administration of their duties. This project will allow the university to add, delete, and modify information about staff, students, courses, clubs, as well as creating and organizing events and competitions. This will make university staff and students' records more secure, efficient, and accessible, ensuring better user experience that will make the client university stand out from competitors.

This project will boost organization by separating courses into degrees and sections, each with an assigned instructor, which enables students to get clear and precise information about what classes they will be taking, which betters ease of use during enrollment. Also, staff with proper clearance will be able to better access information for each classes' students, as well as being able to see data related to each class, like students' grades, or the failure percentage of a class. This allows staff to better manage their syllabus, or to better judge professors' performance.

Furthermore, this project aims to connect the client university to others like it around the world, allowing its students to expand their horizons through exchange programs. Information about foreign universities and the exchange process will be organized and available for students to see, allowing them to plan their studies abroad more easily, as well as providing international students with an easier transition into the university.

To enhance student participation in activities and sports, this Database Management System will store information about events and competitions the university participates in, as well as information about teams and clubs. This will help the university when deciding on what to invest, as they will have data on what students like to spend time on, and what activities have the most attendees.

# **Section II:** Use Cases

---

1. **Use Case:** Student Class Enrollment

**Actors:** Lewis (Student), Class

**Description:** Lewis has returned from summer vacation and is starting college in a few weeks. He needs to enroll in the classes he must take the next semester, but he needs to find a schedule that works for him. Last year, he failed a course, so he needs to add and accommodate this course into the ones he is supposed to take this semester, and this year he plans on working to earn some extra money. However, he is not sure if he will be able to pass all his courses if he is overloaded with work. He is planning on learning which courses are easiest and leaving them for a later semester where he has more time: only enrolling in the hardest ones, and the course he already failed.

The College Database System we are designing will help Lewis find information about the courses the university offers, organized by degrees, year, and semester when they are taught. He will be able to access the course syllabus, as well as the professors that teach that class, and the percentage of passing students. He can enroll in these classes and drop them, and the database will keep an updated list of his courses, and their enrollment status.

2. **Use Case:** Updating Grades

**Actors:** Hirva (Professor), Class, Student, Exam

**Description:** Hirva teaches several classes in her university and has just finished grading the midterms from one of her classes. She is tired, and wants to upload her students' grades as quickly as possible and without any complications. She doesn't know the names of all her students, so she needs a way to filter her students and have all the information organized. Using the old university's system, she has made several mistakes before, so she hopes to save herself from students requesting exam revisions when they receive unexpected grades.

Our Database System is designed to make this process easier, by having exams and students sorted by classes, so that professors have a list of students for each exam for each class, so that grades can be updated without unnecessary complications.

3. **Use Case:** Head of Department adds a Foreign Exchange University

**Actors:** Brandon (Head of Department), University, Course

**Description:** Brandon, the head of his university's Computer Science department, has noticed that the COVID-19 crisis is almost at its end in some countries outside the US. Due to the pandemic, the university had to cancel its exchange programs, and the list of available foreign universities had to be severely modified. For 2022, his superiors have decided to open the university up to foreign exchange students and allow students to study abroad. After meetings and negotiations, agreements have been made with other universities to create exchange programs. Brandon wants to add these universities to the list of universities available with their corresponding information, so that students can learn about the new foreign exchange program.

The College Database System aids in updating this information by allowing users with appropriate clearance to manage these tables, organizing them by countries, and containing pertinent information such as location, number of available admissions, cost, exchange duration, and more. Once edited, this information can be viewed by students and can be updated at any given time.

4. **Use Case:** Student wants to update personal information

**Actors:** Jorge (Student), Address, Phone number

**Description:** Jorge had to change his phone number and has changed apartment this year. The university has told him that it is necessary for him to update his information in the university's records. He doesn't want to make a mistake with this, because last time he got lost during the process of updating his data and wrote down the wrong address. He expects to be done quickly with this task, as it is very simple.

After logging in to his account, Jorge enters his profile, where he has his personal information listed. He edits the fields he wishes to change and saves his changes before moving on to other fields he wished to change. The database is updated in real time, and other users with permission can look at his profile to get information on that student.

5. **Use Case:** Head of Department wants to evaluate professor

**Actors:** Lisa (Head of Dept), Professor, Course

**Description:** Lisa is the Head of the Economics Department at the university she works at. She has heard several complaints from students about a new professor who is extremely demanding

in a class which is not supposed to be too challenging, so she has decided to investigate more about this. She thought that a good way to evaluate the harshness of this new professor was to look at the proportion of failing students in this professor's classes compared to other professors, and previous years' classes.

Our College Database System allows Lisa to view the classes this professor teaches, and the percentage of failing students. She can also access other professors' classes in the same subject to compare the data and look up previous years' classes to see the failure rates.

6. **Use Case:** Student wants to learn about clubs & events

**Actors:** Jeffrey (Student), Club, Event

**Description:** Jeffrey is an incoming freshman in his new university who wants to get to know people and build a social circle for his college experience. He has heard that his university has different fraternities, clubs, and events where he can socialize and make new friends. However, he has heard that some of these are not recognized by the university, so he only wants to see which are serious and official.

Jeffrey can find the information he is looking for in the university's Database Management System. There he can find listed the fraternities, sororities, clubs, and events he can partake in. He can find information about these aspects of university life, but to enlist in one of these he needs to be admitted by a user with the necessary permissions to edit the content of these groups.

7. **Use Case:** Rector adds Degree

**Actors:** Sarah (Rector), Degree, Course

**Description:** Sarah is the rector of her university, and as the highest authority in her institution she can decide which degrees are offered at her university. After some meetings, the university board has decided to stop offering a degree, and to create new one. Sarah then uses her credentials to log in to the database system, and in the degrees section she closes the degree she wants to close. Future students will not be able to choose this degree option, and courses which are unique to this degree will be deleted for future classes. Then, she adds the new degree to the database, with its corresponding courses. Current students are not allowed to enroll in this degree, but in the next academic session, they will be.

# **Section III:** Database Requirements

---

1. **User**

a. A user is either a student, a professor, a head of department, a rector, or an administrator.

b. A user shall attend many events.

c. Only users who are students may be club, frat, or team members.

d. A user shall have at least one set of credentials.

e. A user can have many gym memberships.

f. Only users who are students can obtain student jobs, insurance, and scholarships.

g. Users who are students can have many jobs.

h. Users who are students can have many insurance contracts.

i. Users who are students can have at most one frat membership.

j. Users who are students can have many club memberships.

k. Users who are students can have many team memberships.

l. Users who are students can obtain many scholarships.

m. Users shall have only one password.

2. **Credentials**

a. Credentials shall be linked to one and only one user.

3. **Student**

a. A student shall enroll in at least one section.

b. A student shall study at least one degree.

c. A student shall be registered by one and only one administrator.

d. A student shall be permitted to perform at least one action.

e. A student shall

4. Student **Job**

a. A job shall belong to many users.

b. Jobs shall be available only to users who are students.

5. **Insurance Contract**

a. An insurance contract can belong to many users.

b. Insurance contracts shall only be available for users who are students.

6. **Scholarship**

a. Scholarships can be awarded to users.

b. Scholarships can only be awarded to users who are students.

7. **Fraternity**

a. A fraternity shall have at least one frat member.

b. Only users who are students can be frat members.

8. **Club**

a. A club shall have at least one member.

b. Only users who are students can be club members.

9. **Team**

a. A team shall have at least one team member.

b. A team shall have at least one match.

c. A team shall compete many competitions.

d. A team shall belong to only one division.

e. Only users who are students can be team members.

10. **Match**

a. A match shall be played by only two different teams.

b. A match shall belong to only one competition.

c. A match shall be hosted at one venue at least.

11. **Competition**

a. A competition shall have more than one team.

b. A competition shall belong to one and only one division.

c. A competition shall have at least one match.

d. A competition shall take place in at least one venue.

12. **Division**

a. A division can have many teams.

b. A division can have many competitions.

13. **Venue**

a. A venue can host many matches.

b. A venue can host many competitions.

14. **Sections**

a. A section shall have at least one student.

b. A section shall have at least one professor.

c. A section shall belong to one and only one course.

d. A section shall have at least one class.

15. **Course**

a. A course shall have at least one section.

b. A course shall be part of at least one degree.

16. **Degree**

a. A degree shall have at least one course.

b. A degree shall belong to at least one department.

c. A degree shall belong to many students.

d. A degree can be equivalent to at least one foreign degree.

17. **Professor**

a. A professor shall teach at least one section.

b. A professor shall be registered by one and only one administrator.

c. A professor will be permitted to perform at least one action.

d. A professor can be the head of at most one department.

18. **Department**

a. A department shall have at least one degree.

b. A department shall have at least one head of department.

20. **Rector**

a. A rector shall manage many departments.

b. A rector shall manage many degrees.

c. A rector shall manage many courses.

d. A rector shall manage many sections.

e. A rector shall be registered by one and only one administrator.

f. A rector shall be permitted to perform at least one action.

21. **Administrator**

a. An administrator shall register all other user types.

b. An administrator shall have permission to perform all the actions in the database, to the whole database.

22. **Actions**

a. An action will be performed by at least one user type.

23. **Class**

a. A class shall be taken by at least one section.

b. A class shall take place in only one classroom.

c. A class shall take place in only one laboratory.

d. A class shall have either a classroom or a laboratory.

24. **Classroom**

a. A classroom can belong to many classes.

25. **Laboratory**

a. A laboratory can belong to many classes.

26. **Event**

a. An event can be attended by many users.

27. **Product**

a. A product can be sold to many stores.

28. **Store**

a. A store shall have at least one product.

29. **Cafeteria**

a. A cafeteria can offer many menus.

30. **Menu**

a. A menu can be found at many cafeterias.

31. **Foreign University**

a. A foreign university can offer many foreign degrees.

32. **Foreign Degree**

a. A foreign degree shall belong to one and only one foreign university.

b. A foreign degree shall be equivalent to at least one degree.

33. **University Gym**

a. University Gyms can have many users.

# **Section IV:** Main Entities, Attributes and Keys

---

When referring to foreign keys, the parameter type is referred to as Entity.Attribute, like class attributes in python.

Also, PK = primary key, and FK = foreign key.

1. **User** (Strong):
    - ***** User_ID: primary key, alphanumeric
    - *Password: alphanumeric
    - *Name: composite, alphanumeric
        1. Name
        2. Surname
    - *Gender: alphanumeric
    - *DOB: date
    - *Age: derived, number

2. **User_phones** (weak):
    - *user: PK/FK alphanumeric
    - *phone: PK/FK number

3. **User_addresses** (weak):
    - *User: PK/FK alphanumeric
    - *Address: PK/FK alphanumeric, composite

4. **Foreign University** (Strong):

*UniversityID: primary key, alphanumeric

*UniversityName: alphanumeric

*Campus: unique key, alphanumeric

*Country: alphanumeric

*City: alphanumeric

*Language: aplhanumeric

*Specialty: alphanumeric.

5. **Club** (Strong):

*ClubID: primary key, alphanumeric

*ClubName: alphanumeric

*Category: alphanumeric

*Description: alphanumeric

*Num_of_Members: number

*Email: alphanumeric

6. **ClubCategory** (weak):

*clubcategory_ID: PK alphanumeric

*club: UK/FK alpanumeric

*category: UK alphanumeric

7. **Event** (Strong):

*Event_ID: primary key, alphanumeric

*EventName: alphanumeric

*Date: primary key, date

*Location: FK, venue

*Category: alphanumeric

*Description: alphanumeric

8. **Division** (Strong)**:**

*DivisionID: primary key, alphanumeric

*DivisionID: alphanumeric

*Sport: alphanumeric

*Num_of_Teams: number

9. **Farternity** (Strong)**:**

*Frat_ID: primary key, alphanumeric

*FratName: alphanumeric

*FoundingDate: date

*Num_of_Members: number

*Date_of_closing: date

10. **FratAddress** (weak):

*frat: PK/FK alphanumeric

*address: PK/FK alphanumeric

11. **Venue** (Strong):

*VenueID: primary key, alphanumeric

*VenueName: alphanumeric

*Capacity: number

12. **Venue_Adress** (weak)**:**

*****Venue: PK/FK alphanumeric

*Address: PK/FK alphanumeric

13. **Classroom** (Strong)**:**

    *Place_ID: primary key, alphanumeric

    *Purpose: alphanumeric

    *Capacity: number

14. **Laboratory** (Strong):

    *Place_ID: primary key, alphanumeric

    *Category: alphanumeric

    *Capacity: number

    *Equipment: alphanumeric

15. **Scholarship** (Strong):

    *Schol_ID: primary key, alphanumeric

    *Company: alphanumeric

    *Num_of_offers: number

    *Sum: number

16. **Insurance Contract** (Strong):

    *Ins_ID: primary key, alphanumeric

    *Company: alphanumeric

    *Money_Available: number

    *Cost_per_month: number

17. **Product** (Strong):

    *Product_ID: primary key, alphanumeric

    *ProdName: alphanumeric

    *Price: number

    *Availability: number

*Weight: number

*Height: number

*Category: alphanumeric

*Description: alphanumeric

18. **Cafeteria** (Strong):

*CafeteriaID: primary key, alphanumeric

*CafeteriaName: alphanumeric

*Opening_hours: composite

1. Hour: number
2. Minute: number

*Closing_hours: composite

1. Hour: number
2. Minute: number

19. **CafeteriaAddress** (weak):

*cafeteria: PK/FK alphanumeric

*address: PK/FK alphanumeric

20. **Gym** (Strong):

*GymID: primary key, alphanumeric

*GymName: alphanumeric

*Capacity: number

*Sports_facilities: alphanumeric

21. **Gym Address** (weak):

*gym: PK/FK alphanumeric

*address: PK/FK alphanumeric

22. **Student Job** (Strong):

*Job_ID: primary key, alphanumeric

*JobName: alphanumeric

*Wage: number

*Employer: alphanumeric

*Requisites: alphanumeric

*Description: alphanumeric

23. **Menu** (Strong):

*Menu_ID: primary key, alphanumeric

*Appetizer: primary key, alphanumeric

*Main_course: primary key, alphanumeric

*Dessert: primary key, alphanumeric

*Drink: alphanumeric

*Price: number

24. **Team** (Weak):

*TeamID: primary key, alphanumeric

*TeamName: alphanumeric

*Division: PK/FK, Division.DivisionID

*Number_of_players: number

25. **Team_member** (Weak):

*User_ID: FK/PK, User.User_ID

*Team: FK/PK, Team.TeamID

***Position: alphanumeric

*Extra_info: alphanumeric

*Number: alphanumeric

*Season: composite, date

    1. Start
    2. End

26. **Frat_member** (Weak):
    *Frat: FK/PK, Fraternity.Frat_ID
    *Member: FK/PK, User.User_ID
    *Status: alphanumeric
    *Date_of_entry: date
    *Date_of_exit: date

27. **Membership** (Weak):
    *Gym: FK/PK, Gym.GymID
    *User: PK/FK, User_ID
    *Type: PK, composite
        1. Price_per_month: number
        2. Type_ID: alphanumeric
        3. Description: alphanumeric
    *Expedition_date: date
28. **Contract** (Weak):
    *User: PK/FK, User.User_ID
    *Job: PK/FK, StudentJob.Job_ID
    *Shift: PK, alphanumeric

29. **Attends** (Weak):
    *User: PK/FK, User.User_ID
    *Event: PK/FK, Event.EventID
    *Ticket: composite
        1. Price: number

2. Tier: alphanumeric

30. **Student** (Weak):
    *User: PK/FK, User.User_ID
    *Student_ID: primary key, alphanumeric
    *AverageGrade: number
    *EnrollmentYear: number

31. **Professor** (Weak):
    *User: PK/FK, User.User_ID
    *Prof_ID: primary key, alphanumeric
    *Salary: number
    *NumberOfSections: number

32. **Head_of_Dept** (Weak):
    *User: PK/FK, User.User_ID
    *Head_ID: primary key, alphanumeric
    *Salary: number
    *NumberOfDepts: number

33. **Rector** (Weak):
    *User: PK/FK, User.User_ID
    *Rector_ID: primary key, alphanumeric
    *Salary: number

34. **Admin** (Weak):
    *User: PK/FK, User.User_ID
    *Admin_ID: primary key, alphanumeric

35. **Permissions** (Weak):

*User: PK/FK, User.User_ID

*Action: PK/FK, alphanumeric

36. **Actions** (Strong):

*Action_ID: primary key, alphanumeric

*Description: alphanumeric

37. **Enrolls** (Weak):

*Student: PK/FK, Student.Student_ID

*Section: PK/FK, Section.Section_ID

*Semester: PK/FK, semester.semester_ID

*Grade: number

38. **Belongs_to** (Weak):

*Department: PK/FK Department.Department_ID

*Degree: PK/FK Degree.Degree_ID

39. **Teaches** (Weak):

*Professor: PK/FK Professor.Professor_ID

*Section: PK/FK Section.Section_ID

*Semester: PK/FK semester.semester_ID

40. **Manages** (Weak):

*Head: PK/FK Professor.Professor_ID

*Department: PK/FK Department.Department_ID

41. **Section** (Weak):

    *Section_ID: primary key, alphanumeric

    *Course: PK/FK Course.Course_ID

    *Number_of_students: number

    *Passing_Students: number

42. **Course** (Weak):

    *Course_ID: primary key, alphanumeric

    *Description: alphanumeric

    *Credits: number

43. **Studies** (Weak):

    *Student: PK/FK Student.Student_ID

    *Degree: PK/FK Degree.Degree_ID

    *Year_Start: number

    *Year_End::number

44. **Is_part_of** (Weak):

    *Course: PK/FK Course.Course_ID

    *Degree: PK/FK Degree.Degree_ID

45. **Degree** (Strong):

    *Degree_ID: primary key, alphanumeric

    *Description: alphanumeric

    *Number_of_Students: derived, number

46. **Department** (Weak):

    *Department_ID: primary key, alphanumeric

    *Number_of_degrees: derived, number

    *Category: alphanumeric

47. **Equivalent** (Weak):

*Degree: PK/FK Degree.Degree_ID

*Foreign_Degree: PK/FK ForeignDegree.ForDeg_ID

48. **ForeignDegree** (strong):

*ForDeg_ID: primary key, numeric

*ForeignUni: PK/FK, ForeignUniversity.ForeignUniversityID

*Description: alphanumeric

49. **Takes** (Weak):

*Section: PK/FK Section.Section_ID

*Class: PK/FK Class.Class_ID

50. **Class** (Weak):

*Class_ID: primary key, alphanumeric

*Time_Slot: primary key, composite

1. Date: date
2. Time: time

51. **Store** (Weak):

*Store_ID: primary key, alphanumeric

*Address: FK/UK address.addressID

52. **Sells** (Weak):

*Store: PK/FK Store.Store_ID

*Product: PK/FK Product.Product_ID

*Stock: number

53. **Offers** (Weak):

*Cafeteria: PK/FK Cafeteria.Cafeteria_ID

*Menu: PK/FK Menu.Menu_ID

*Time Frame: composite

1. DateStart: date
2. DateEnd: date

54. **Club_member** (Weak):

*User_ID: FK/PK, User.User_ID

*Club: FK/PK, Club.Club_ID

***Role: alphanumeric

*Extra_info: alphanumeric

55. **Winner** (Weak):

*Student: PK/FK, Student.Student_ID

*Scholarship: PK/FK Scholarship. Schol_ID

56. **Plan** (Weak):

*Student: PK/FK, Student.Student_ID

*Insurance: PK/FK, Insurance.Insurance_ID

*Date_Bought: date

57. **Takes_Class_In** (Weak):

*Class: PK/FK Class.Class_ID

*Place: PK/FK Classroom.Place_ID

58. **Competition** (Weak):

*Competition_ID: primary key, alphanumeric

*Division: PK/FK, Division.Division_ID

*Number_of_teams: number

*Reward: number

59. **Competes** (Weak):

*Competition: PK/FK Competition.Competition_ID

*Team: PK/FK Team.Team_ID

*Season: Primary key, number

*Win: Boolean

60. **Match** (Weak):

*Match_ID: primary key, alphanumeric

*Team1: PK/FK Team.Team_ID

*Team2: PK/FK Team.Team_ID

*Date:  PK, date

61. **Hosts** (Weak):

*Match: PK/FK Match.Match_ID

*Venue: PK/FK Venue.Venue_ID

62. **Address** (strong):

* Country: PKalphanumeric

*City: PK alphanumeric

*Street: PK alphanumeric

*Apartment: (optional) alphanumeric

* Postal code: PK number

63. **Phone** (strong):

*Prefix: number

*Number: PK number

64. **Semester** (strong):

*Semester_ID: PK alphanumeric

*year: UK year

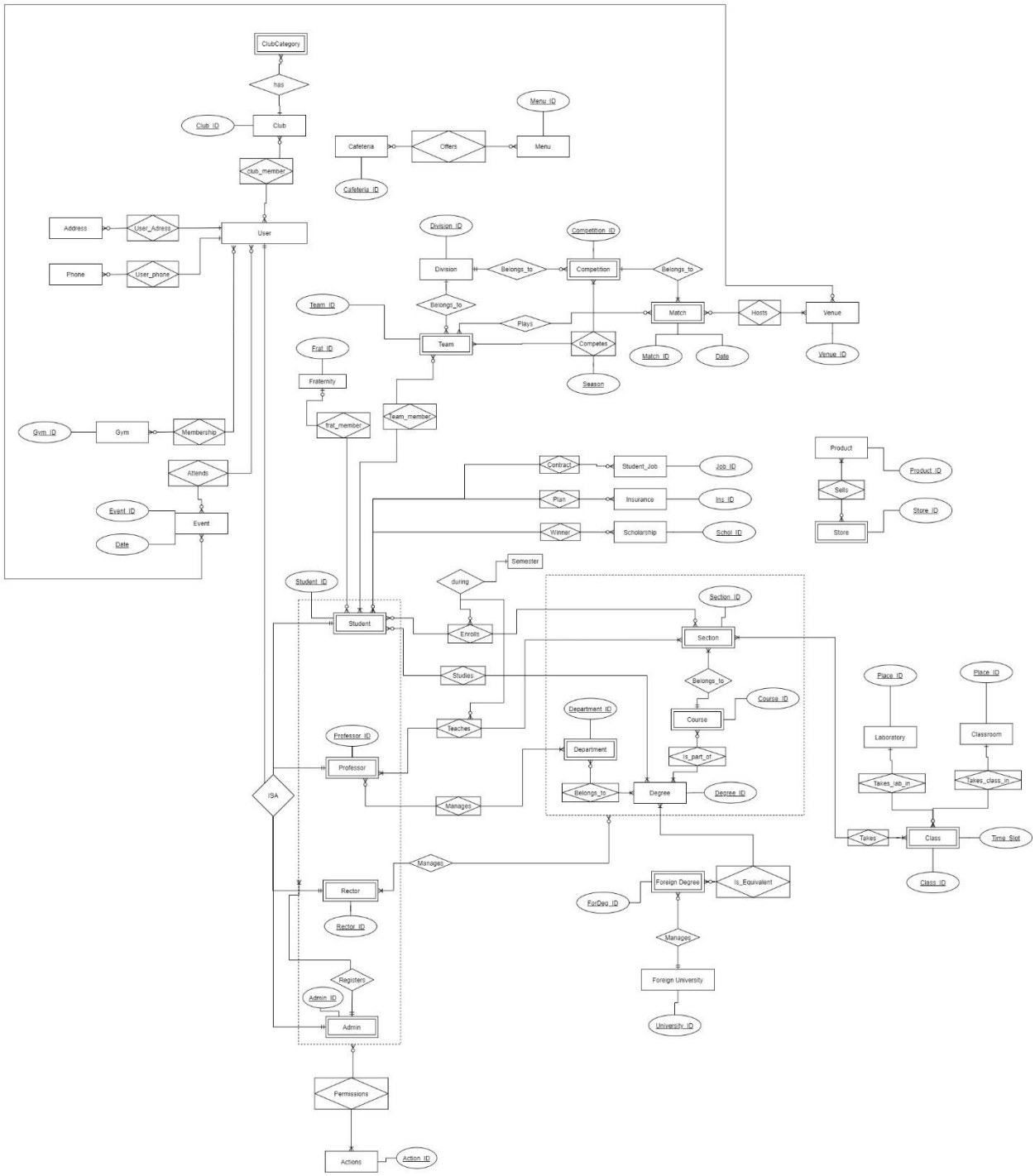
*period: UK alphanumeric

65.**Takes_Lab_in** (weak):

*Class: PK/FK Class.Class_ID

*Place: PK/FK Laboratory.Place_ID

# Section V: Entity Relationship Diagram



https://drive.google.com/file/d/1PwVtBgv7G6svKVo8WOUL1_3c-dyweL5R/view?usp=sharing

# Section VI: Testing Table

---

| Rule | Entity 1 | Relation | Entity 2 | Cardinality | Pass/Fail |
|---|---|---|---|---|---|
| | | | | | |
| 1.a | User | ISA | Student | 1:1 | PASS |
| 1.a | User | ISA | Professor | 1:1 | PASS |
| 1.a | User | ISA | Head of Department | 1:1 | PASS |
| 1.a | User | ISA | Rector | 1:1 | PASS |
| 1.a | User | ISA | Admin | 1:1 | PASS |
| 1.b | User | Attend | Event | M:N | PASS |
| 1.c | Student | Is_member | Club | M:N | FAIL |
| 1.c | Student | Is_member | Team | M:N | FAIL |
| 1.c | Student | Is_member | Frat | M:1 | FAIL |
| 1.d | User | Have | Credentials | 1:M | PASS |
| 1.e | User | Membership | UniversityGym | M:N | PASS |
| 1.g | Student | Obtains | StudentJob | M:N | FAIL |
| 1.h | Student | Obtains | Insurance | M:N | FAIL |
| 1.i | Student | Is_member | Frat | M:1 | FAIL |
| 1.j | Student | Is_member | Club | M:N | FAIL |
| 1.k | Student | Is_member | Team | M:N | FAIL |
| 1.l | Student | Obtain | Scholarship | M:N | FAIL |
| 2.a | Credentials | Have | User | M:1 | PASS |
| 3.a | Student | Enroll | Section | M:N | FAIL |
| 3.b | Student | Belongs_to | Degree | M:N | PASS |
| 3.c | Student | Registered_by | Administrator | M:1 | PASS |
| 3.d | Student | Permission | Action | 1:M | PASS |
| 4.a | StudentJob | Obtain | Student | N:M | FAIL |
| 4.b | StudentJob | Obtain | Student | N:M | FAIL |
| 5.a | Insurance | Obtain | Student | N:M | FAIL |
| 5.b | Insurance | Obtain | Student | N:M | FAIL |
| 6.a | Scholarship | Obtain | Student | N:M | FAIL |
| 6.b | Scholarship | Obtain | Student | N:M | FAIL |
| 7.a | Frat | Has | Student | 1:M | FAIL |
| 7.b | Frat | Has | Student | 1:M | FAIL |
| 8.a | Club | Has | Student | N:M | FAIL |
| 8.b | Club | Has | Student | N:M | FAIL |
| 9.a | Team | Has | Student | N:M | FAIL |
| 9.b | Team | Play | Match | 2:M | PASS |

| | | | | | |
|---|---|---|---|---|---|
| 9.c | Team | Compete | Competition | N:M | PASS |
| 9.d | Team | Belongs_to | Division | M:1 | PASS |
| 9.e | Team | Has | Student | N:M | FAIL |
| 10.a | Match | Played_by | Team | M:2 | PASS |
| 10.b | Match | Belongs_to | Competition | M:1 | PASS |
| 10.c | Match | Hosted_at | Venue | M:1 | PASS |
| 11.a | Competition | Has | Team | M:N | PASS |
| 11.c | Competition | Has | Match | 1:M | PASS |
| 11.d | Competition | Hosted_at | Venue | M:N | PASS |
| 12.a | Division | Have | Team | 1:M | FAIL |
| 12.b | Division | Have | Competition | 1:M | PASS |
| 13.a | Venue | Host | Match | 1:M | PASS |
| 13.b | Venue | Host | Competition | M:N | PASS |
| 14.a | Section | Has | Student | M:N | FAIL |
| 14.b | Section | Has | Professor | M:1 | FAIL |
| 14.c | Section | Belongs_to | Course | M:1 | PASS |
| 14.d | Section | Has | Class | 1:M | FAIL |
| 15.a | Course | Has | Section | 1:M | FAIL |
| 15.b | Course | Belongs_to | Degree | M:N | FAIL |
| 16.a | Degree | Has | Course | M:N | PASS |
| 16.b | Degree | Belongs_to | Department | M:N | PASS |
| 16.c | Degree | Belongs_to | Student | M:N | FAIL |
| 16.d | Degree | Be_equivalent | Foreign Degree | M:N | PASS |
| 17.a | Professor | Teach | Section | 1:M | FAIL |
| 17.b | Professor | Be_registered | Administrator | M:1 | PASS |
| 17.c | Professor | Permission | Action | M:N | PASS |
| 18.a | Department | Has | Degree | M:N | PASS |
| 18.b | Department | Has | Head_of_ department | M:N | PASS |
| 19.a | Head of Dept | Belongs_to | Department | M:N | PASS |
| 19.b | Head of Dept | Be_registered | Administrator | M:1 | PASS |
| 19.c | Head of Dept | Permission | Action | M:N | PASS |
| 20.a | Rector | Manage | Department | M:N | PASS |
| 20.b | Rector | Manage | Degree | M:N | PASS |
| 20.c | Rector | Manage | Course | M:N | PASS |
| 20.d | Rector | Manage | Section | M:N | PASS |
| 20.e | Rector | Be_registered | Administrator | M:1 | PASS |
| 20.f | Rector | Permission | Action | M:N | PASS |
| 21.a | Administrator | Register | User | 1:M | PASS |
| 21.b | Administrator | Permission | Action | 1:M | PASS |
| 22.a | Action | Performed_by | User | M:N | PASS |

| | | | | | |
|---|---|---|---|---|---|
| 23.a | Class | Taken_by | Section | M:1 | PASS |
| 23.b | Class | Take_place | Classroom | M:N | PASS |
| 23.b | Class | Take_place | Laboratory | M:N | PASS |
| 24.a | Classroom | Belongs_to | Class | 1:M | PASS |
| 25.a | Laboratory | Belong_to | Class | 1:M | PASS |
| 26.a | Event | Attended_by | User | M:N | PASS |
| 27.a | Product | Sold_in | Store | M:N | PASS |
| 28.a | Store | Have | Product | M:N | PASS |
| 29.a | Cafeteria | Offer | Menu | M:N | PASS |
| 30.a | Menu | Be_found | Cafeteria | M:N | PASS |
| 31.a | Foreign University | Offer | Foreign Degree | 1:N | PASS |
| 32.a | Foreign Degree | Belongs_to | Foreign University | M:1 | PASS |
| 32.b | Foreign Degree | Be_equivalent | Degree | M:N | PASS |
| 33.a | University Gym | Have | User | M:N | PASS |

| Test | Error Description |
|---|---|
| 1.c<br>1.g<br>1.h<br>1.i<br>1.k<br>1.l<br>4.a<br>4.b<br>5.a<br>5.b<br>6.a<br>6.b<br>7.a<br>7.b<br>8.a<br>8.b<br>9.a<br>9.e | Student is weak, so this makes all relations to jobs, scholarships, insurance, clubs, fraternities, and teams obligatory. We want these relations to be optional for students. |
| 3.a<br>14.a | Students can’t enroll into several different sections and details like semester are missing. |
| 12.a | Divisions can’t have no teams from the home university, but divisions are external to the university. |
| 14.b<br>17.a | Professors can’t teach several sections. More details are needed, like semester. |

| | |
|---|---|
| 14.d | Section can only have one class, and we want the section to have many classes. |
| 15.a | Course doesn’t support many sections. More details are needed for this relation, as a course can have many sections and sections can belong to many courses. |
| 15.b | Degree doesn’t support many courses. More details are needed for this relation, as a degree can have many courses and courses can belong to many degrees. |
| 16.c | Student doesn’t support many degrees. More details are needed for this relation, as a degree can have many students and students can belong to many degrees. |

# Section VII: Database Model / EER

| Table | FK | ON UPDATE | ON DELETE | Comment |
|---|---|---|---|---|
| CafeteriaAddress | Cafeteria | CASCADE | CASCADE | CafeteriaAddress gets deleted if cafeteria gets deleted. |
| CafeteriaAddress | Address | CASCADE | CASCADE | CafeteriaAddress gets deleted if address is deleted. |
| ClubMember | User | CASCADE | CASCADE | ClubMember gets deleted if user gets deleted. |
| ClubMember | Club | CASCADE | CASCADE | ClubMember gets deleted if club gets deleted. |
| Department | Professor | CASCADE | SETNULL | Department head can be null if professor gets deleted. |
| GymMembership | Gym | CASCADE | CASCADE | GymMembership gets deleted if gym gets deleted. |
| GymMembership | User | CASCADE | CASCADE | GymMembership gets deleted if usergets deleted |
| Professor | User | CASCADE | CASCADE | Professor gets deleted if user gets deleted. |
| Store | Address | CASCADE | SETNULL | A store can have no address. |
| TakesLabIn | Class | CASCADE | CASCADE | If a class is deleted, takeslabin gets deleted. |
| TakesLabIn | Laboratory | CASCADE | CASCADE | If a laboratory is deleted, takeslabin is deleted. |
| UserPhones | User | CASCADE | CASCADE | Userphones cant exist without user |
| UserPhones | Phones | CASCADE | CASCADE | Userphones cant exist without phone |
| Competes | Team | CASCADE | CASCADE | Competes cant exist without Team |
| Competes | Competition | CASCADE | CASCADE | Competes cant exists without Competition |
| FratAddress | Fraternity | CASCADE | CASCADE | FratAddress cant exist without Fraternity |
| FratAddress | Address | CASCADE | CASCADE | FratAddress cant exist without Address |
| Hosting | Match1 | CASCADE | CASCADE | Hosting cant exist without Match1 |
| Hosting | Venue | CASCADE | CASCADE | Hosting cant exist without a venue. |
| Offers | Cafeteria | CASCADE | CASCADE | Offers cant exist without cafeteria |
| Offers | Menu | CASCADE | CASCADE | Offers cant exist without menu |
| Rector | User | CASCADE | CASCADE | Rectors cant exists without a user. |
| Student | User | CASCADE | CASCADE | Students cant exist without a user. |
| Teaches | Professor | CASCADE | CASCADE | Teaches cant exist without a professor |
| Teaches | Section | CASCADE | CASCADE | Teaches cant exist without a section |
| Teaches | Semester | CASCADE | CASCADE | Teaches cant exist without a semester |
| Competition | Division | CASCADE | RESTRICT | Division cant be deleted. |
| Enrolls | Student | CASCADE | CASCADE | Enrolls cant exist without a student. |
| Enrolls | Section | CASCADE | CASCADE | Enrolls cant exist without a section. |

| | | | | |
|---|---|---|---|---|
| Enrolls | Semester | CASCADE | RESTRICT | Enrolls cant exist without a semester. |
| Permissions | User | CASCADE | CASCADE | Permissions needs a user. |
| Permissions | Action | CASCADE | CASCADE | Permissions needs an action. |
| Team | Division | CASCADE | RESTRICT | Team needs a division. |
| VenueAddress | Venue | CASCADE | CASCADE | VenueAddress needs a venue. |
| VenueAddress | Address | CASCADE | CASCADE | venueAddress needs an Address. |
| Admin | User | CASCADE | CASCADE | Admin needs a user. |
| Contract | Student | CASCADE | CASCADE | Contracts need a student. |
| Contract | StudentJob | CASCADE | CASCADE | Contract needs a StudentJob. |
| Equivalent | Degree | CASCADE | CASCADE | Equivalent needs a degree. |
| Equivalent | ForeignDegree | CASCADE | CASCADE | Equivalent needs a Foreigndegree. |
| FratMember | Fraternity | CASCADE | CASCADE | FratMember needs a Fraternity. |
| FratMember | Student | CASCADE | CASCADE | FratMember needs a student. |
| IsPartOf | Course | CASCADE | CASCADE | IsPartOf needs a Course. |
| IsPartOf | Degree | CASCADE | CASCADE | IsPartOf needs a degree. |
| Section | Course | CASCADE | CASCADE | Section needs a course. |
| Studies | Student | CASCADE | CASCADE | Studies needs a student. |
| Studies | Degree | CASCADE | RESTRICT | Studies needs a Degree. |
| TeamMember | Student | CASCADE | CASCADE | TeaMember needs a student. |
| TeamMember | Team | CASCADE | CASCADE | TeamMember needs a team. |
| Winner | Student | CASCADE | CASCADE | Winner needs a student. |
| Winner | Scholarship | CASCADE | CASCADE | Winner needs a scholarship. |
| Attends | User | CASCADE | CASCADE | Attends needs user. |
| Attends | Event | CASCADE | CASCADE | Attends needs an event. |
| Event | Venue | CASCADE | SET NULL | Event doesn’t need a Venue. |
| Plan | Student | CASCADE | CASCADE | Plan needs a student. |
| Plan | Insurance | CASCADE | CASCADE | Plan needs Insurance. |
| Sells | Store | CASCADE | CASCADE | Sells needs a store. |
| Sells | Product | CASCADE | CASCADE | Sells needs a product. |
| Takes | Section | CASCADE | CASCADE | Takes needs a section. |
| Takes | Class | CASCADE | CASCADE | Takes needs a class. |
| BelongsTo | Department | CASCADE | CASCADE | BelongsTo needs a department. |
| BelongsTo | Degree | CASCADE | CASCADE | BelongsTo needs a degree. |
| ClubCategory | Club | CASCADE | CASCADE | ClubCategory needs a club. |
| ForeignDegree | ForeignUniversity | CASCADE | CASCADE | ForeignDegree needs a foreignUni |
| GymAddress | Gym | CASCADE | CASCADE | GymAddress needs a Gym. |
| GymAddress | Address | CASCADE | CASCADE | GymAddress needs an address. |
| Match1 | Team | CASCADE | CASCADE | Match1 needs two teams. |
| Match1 | Team | CASCADE | CASCADE | Match1 needs two teams. |
| Match1 | Competition | CASCADE | RESTRICT | Match1 needs a competition. |
| TakesPlaceIn | Class | CASCADE | CASCADE | TakesPlaceIn needs a class. |
| TakesPlaceIn | Classroom | CASCADE | CASCADE | Takesplacein needs a classroom. |
| UserAddress | User | CASCADE | CASCADE | UserAddress needs a User. |
| UserAddress | Address | CASCADE | CASCADE | UserAddress needs an Address. |

# Section XI: Database Model / EER

---

All my tests passed, except for some syntax errors (I considered those to be trivial so I corrected them).

| Entity | SQL Query | Pass/Fail | Error Description | Possible solution |
|---|---|---|---|---|
| Club | UPDATE | PASS | N/A | N/A |
| Club | DELETE | PASS | N/A | N/A |
| Cafeteria | UPDATE | PASS | N/A | N/A |
| Cafeteria | DELETE | PASS | N/A | N/A |
| Menu | UPDATE | PASS | N/A | N/A |
| Menu | DELETE | PASS | N/A | N/A |
| User | UPDATE | PASS | N/A | N/A |
| User | DELETE | PASS | N/A | N/A |
| Address | UPDATE | PASS | N/A | N/A |
| Address | DELETE | PASS | N/A | N/A |
| Phone | UPDATE | PASS | N/A | N/A |
| Phone | DELETE | PASS | N/A | N/A |
| Gym | UPDATE | PASS | N/A | N/A |
| Gym | DELETE | PASS | N/A | N/A |
| Division | UPDATE | PASS | N/A | N/A |
| Division | DELETE | PASS | N/A | N/A |
| Venue | UPDATE | PASS | N/A | N/A |
| Venue | DELETE | PASS | N/A | N/A |
| Fraternity | UPDATE | PASS | N/A | N/A |
| Fraternity | DELETE | PASS | N/A | N/A |
| StudentJob | UPDATE | PASS | N/A | N/A |
| StudentJob | DELETE | PASS | N/A | N/A |
| Insurance | UPDATE | PASS | N/A | N/A |
| Insurance | DELETE | PASS | N/A | N/A |
| Scholarship | UPDATE | PASS | N/A | N/A |
| Scholarship | DELETE | PASS | N/A | N/A |
| Product | UPDATE | PASS | N/A | N/A |
| Product | DELETE | PASS | N/A | N/A |
| Degree | UPDATE | PASS | N/A | N/A |
| Degree | DELETE | PASS | N/A | N/A |
| Laboratory | UPDATE | PASS | N/A | N/A |
| Laboratory | DELETE | PASS | N/A | N/A |
| Classroom | UPDATE | PASS | N/A | N/A |
| Classroom | DELETE | PASS | N/A | N/A |
| ForeignUniversity | UPDATE | PASS | N/A | N/A |
| ForeignUniversity | DELETE | PASS | N/A | N/A |
| Action | UPDATE | PASS | N/A | N/A |
| Action | DELETE | PASS | N/A | N/A |

| | | | | |
|---|---|---|---|---|
| Event | UPDATE | PASS | N/A | N/A |
| Event | DELETE | PASS | N/A | N/A |