Project introduction

This database represents a typical university system. It stores information about classes, students, faculty, and some additional services such as amenities, student clubs, and advisors. The topic for this database was chosen so that we could better understand the details and relationships that are part of a well-organized, well-functioning university system.

The database can be used in the following ways:

- Faculty can use it to find which courses they are teaching and which students are helping them with projects.
- Students can use it to find their advisor, search for courses to register for, and find their current grades.
- Management can use it to record employment information and see which staff members are in charge of which amenities, which TAs work for which classes, and which professors teach which courses.
- Department managers can use it to officially recognize student clubs with faculty sponsorships and funding based on attendance. Each department can also use it to create new courses and offer sections of them each semester.
- Faculty and students can use it to find their TA for each course section they participate in.

Some important aspects of the database's design include:

- Students can be employed as university staff.
- All amenities have at least one staff member maintaining it.
- All clubs have one professor sponsoring it, but this sponsor can be changed anytime.
- Professors can have separate sets of students related to them through either course sections they teach or projects they work on.

Requirements Analysis

- The database stores all the courses taught at the University. Each course has a unique CRN, the database also stores the name of the course, meeting time and location of each course. Each course is divided up into sections and teaching assistants are assigned to each section.
- Each section has its section number and the course CRN associated with

it. The database also stores the office hour and location of the TA assigned to that section. Every section has only one teaching assistant and a teaching assistant can be assigned to only one section.

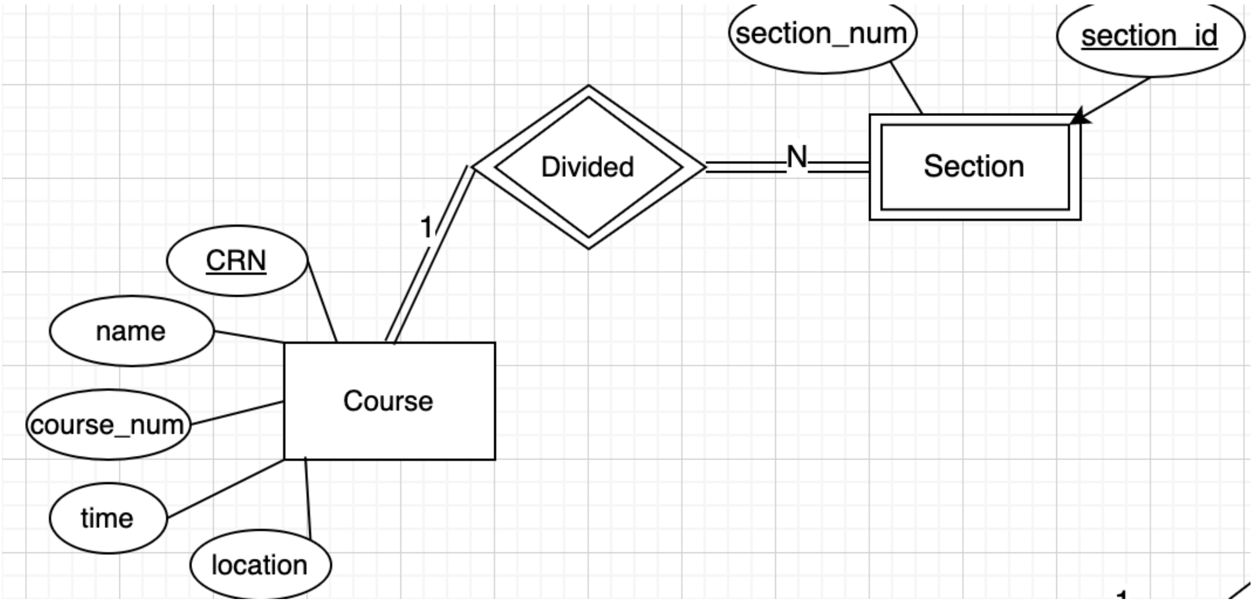
- A course can be taught by multiple professors and one professor can teach many courses.
- A professor is identified by their employee number. The database also stores the professor's name, SSN, email, and hiring date.
- A student is identified by their student ID number. The database also stores their name, SSN, birthday, address, and email. Every professor belongs to a department. Every student has to have an advisor, but it is possible for the advisor to change.
- An advisor is identified by their employee number. The database also stores their name, email, and SSN. Every advisor has to have at least one student assigned to them.
- Every project has a unique name. The database also stores a short description of the project. Every project has to have a lead professor and can have multiple students assisting on the project.
- Each department has its unique name. There is at least one professor in each department and every department has to offer at least one course.
- Every Club has a unique name and it is run by students. The database also stores the budget of each club and the president of the club. Many students can be a part of many clubs. Each club also has one department associated with it.
- A club can sometimes reserve the amenities on campus for activities or meetings. The database records the time and room for the reservation.
- Every amenity on campus has a unique name, the database also stores its address. Each amenity has at least one staff member tasked with maintaining it.
- A staff is identified by their employee ID, the database also stores their name, email address and salary. A student can work as a staff member.

- The database is able to do queries such as listing the names of professors who teach more than one class as well as a list of classes taught by a professor.
- The database could register a student into a section.
- Users can also replace a course stored in the database with a new one.
- The database shows all the sections of a course and student's attendance

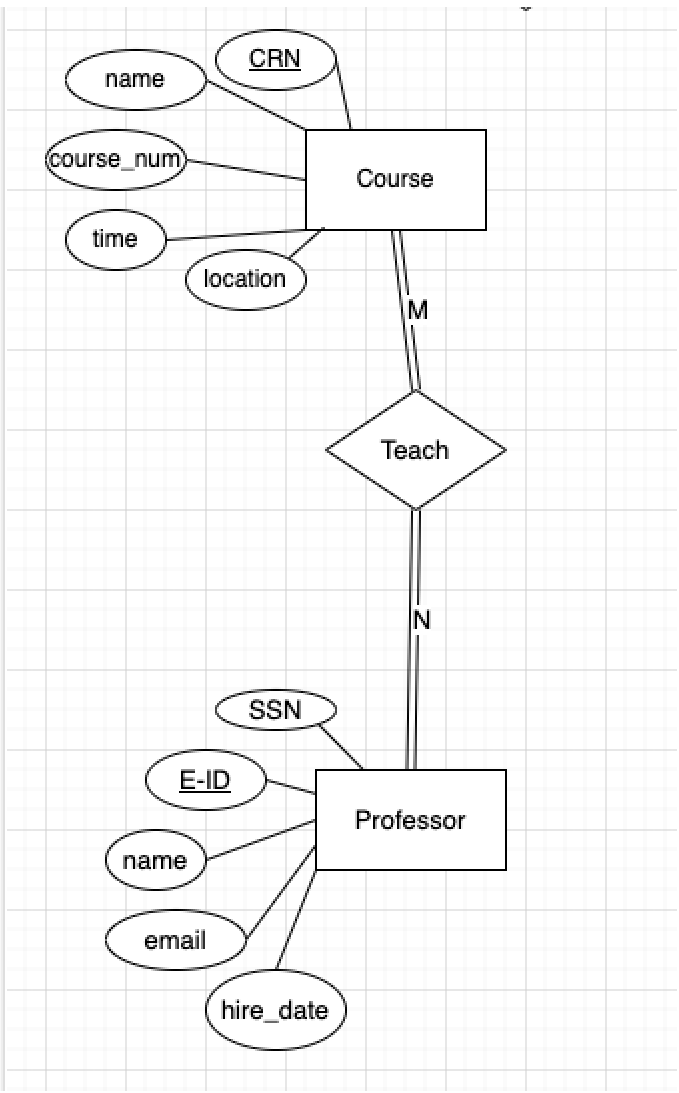
for each section.

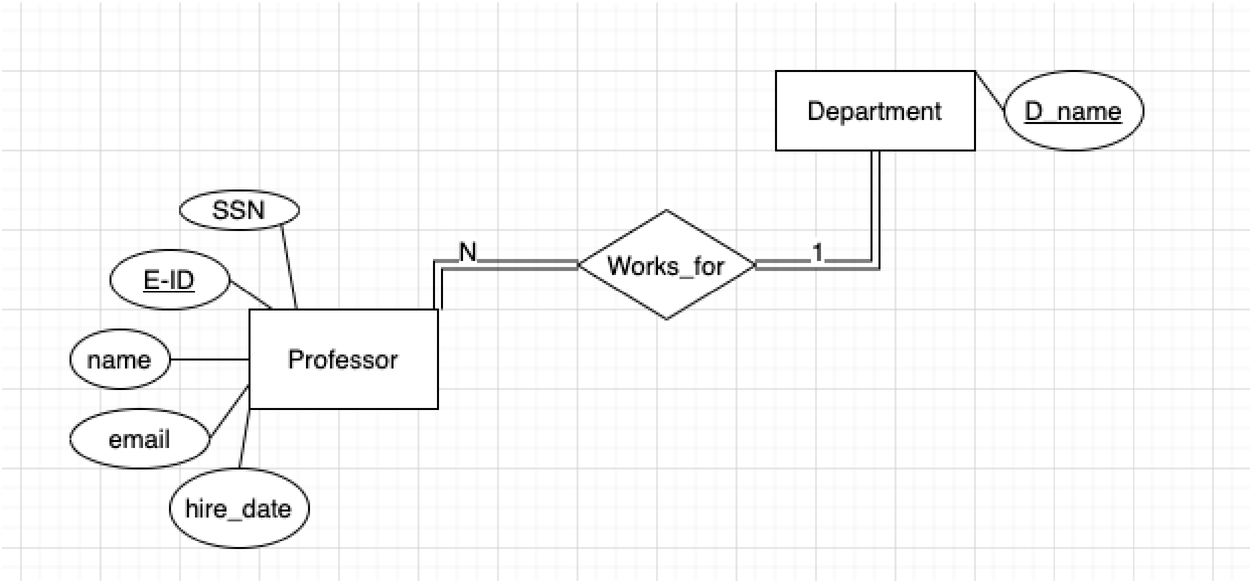
- The sponsor of a club can be updated in the database
- Every amenity on campus has to have one maintenance staff, attempting to remove the staff information should fail.
- Users can reserve an amenity on campus for a club event.
- Given the name of the club, the database is able to generate a list of students belonging to that club.
- Given the name of an amenity, the database is able to generate all the reservation records related to that amenity.
- Given the department name, the database is able to generate a list of projects under the department as well as its leading professor and a list of students assisting it.
- Users can change the leading professor of a project, but simply attempting to remove the leading professor would fail.

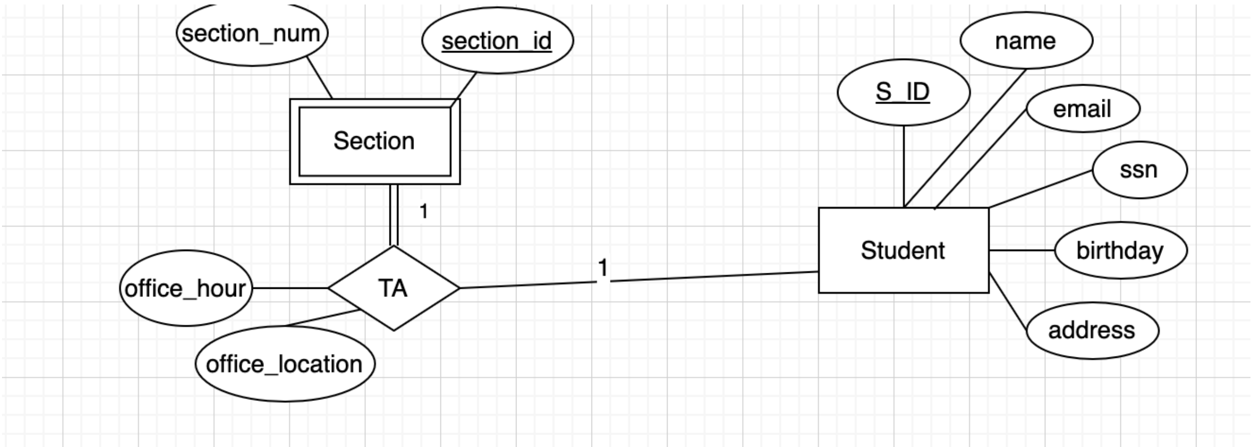
ER Model

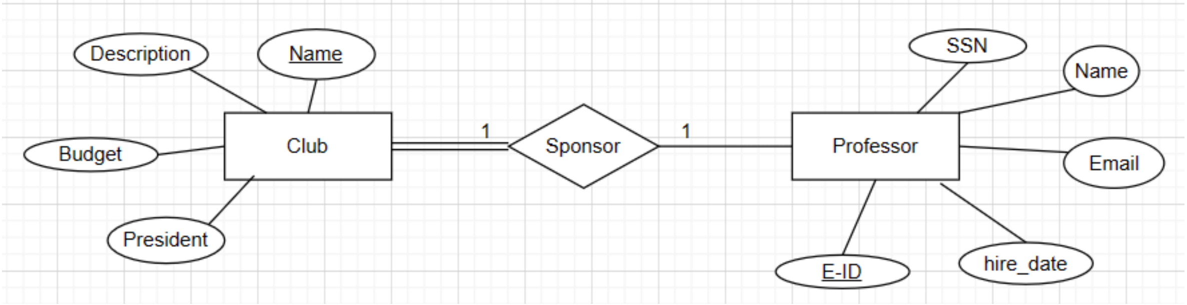


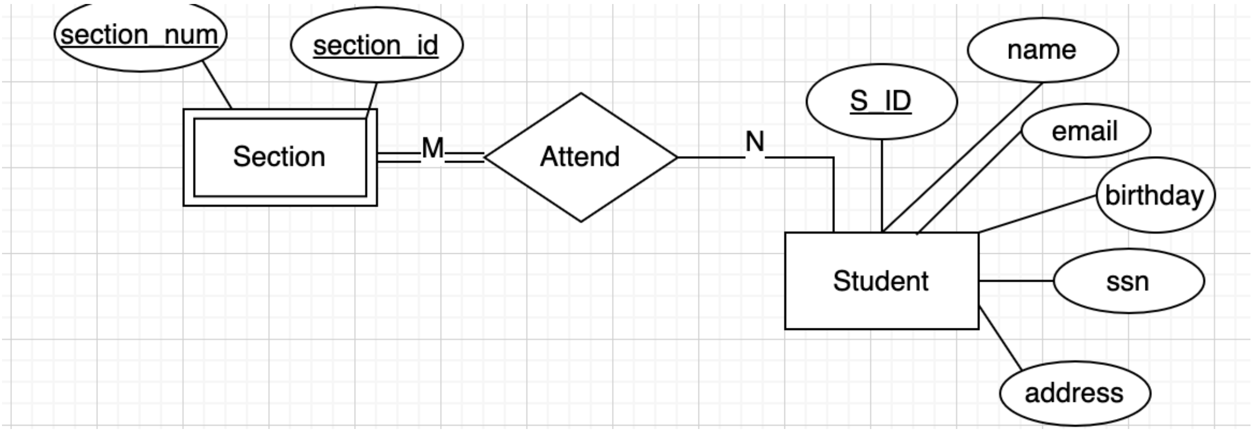
-

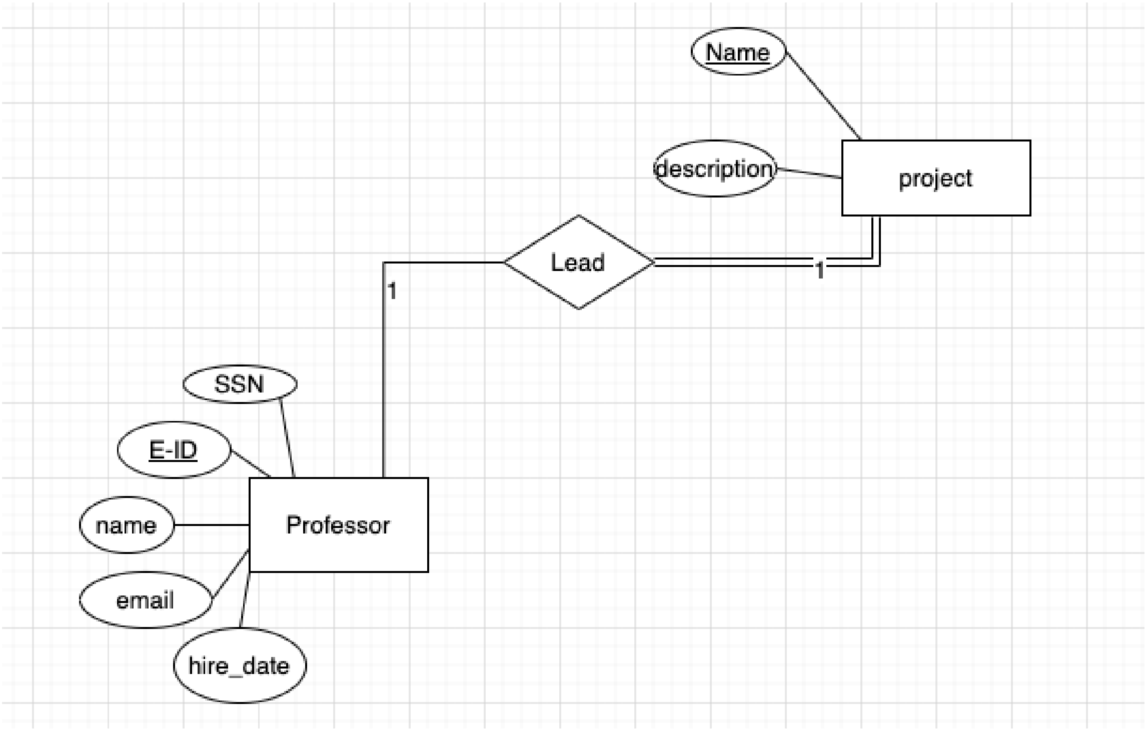
- 
CRN
name
course_num
Course
time
location
M
Teach
N
SSN
E-ID
name
Professor
email
hire_date

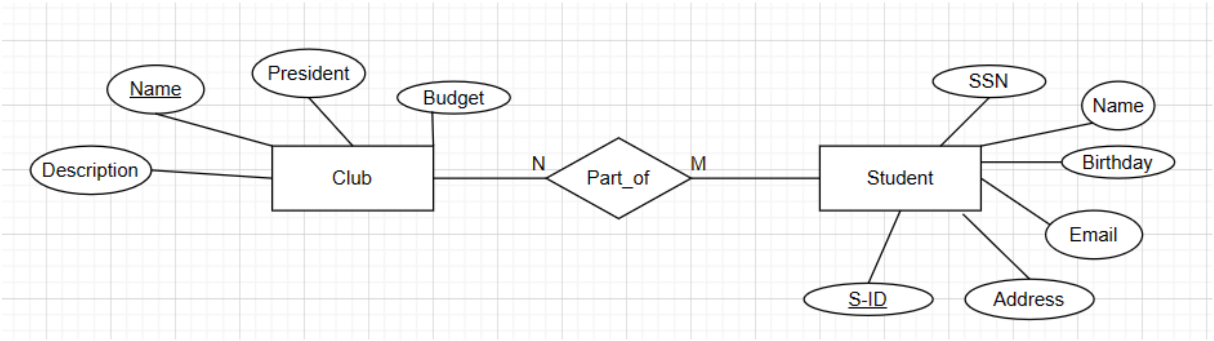
- 
Department
D_name
SSN
E-ID
N
Works_for
1
name
Professor
email
hire_date

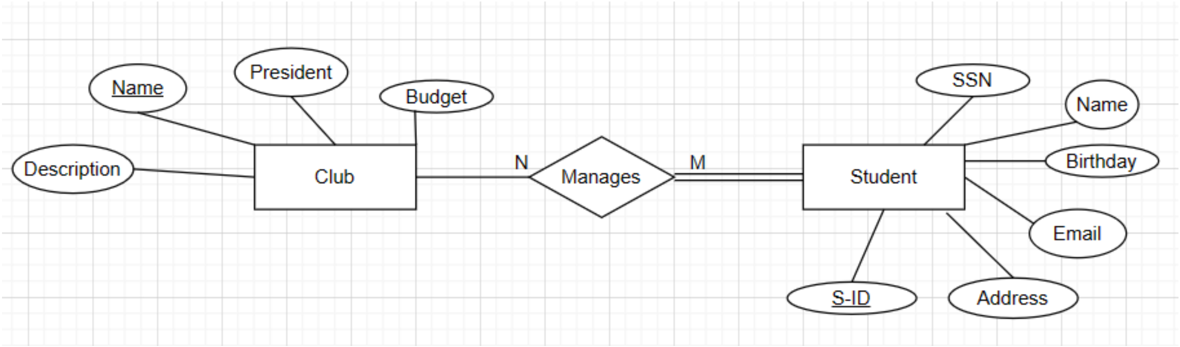
- 
section_num
section_id
Section
1
office_hour
TA
1
office_location
S_ID
name
email
ssn
Student
birthday
address

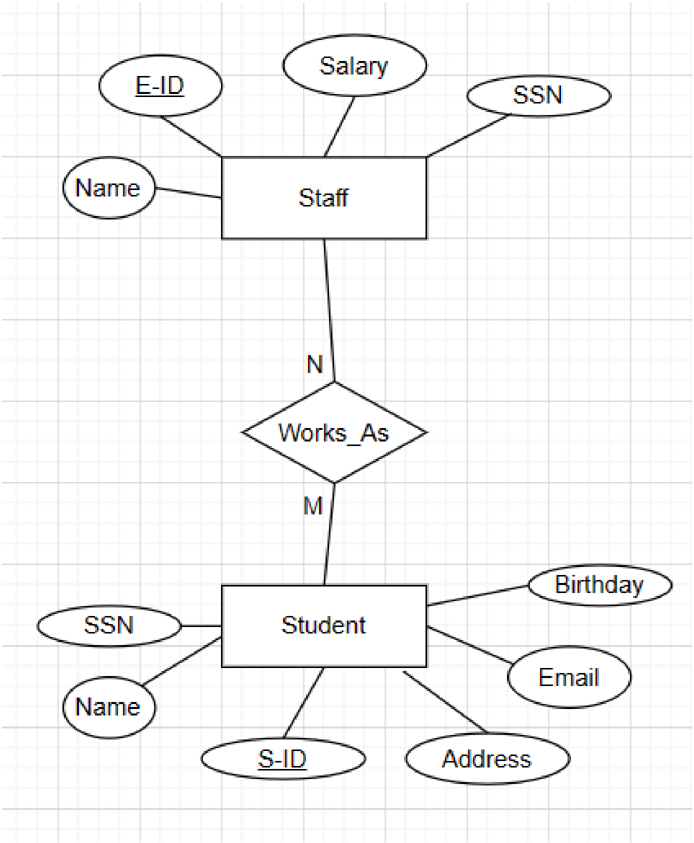
- 
Description
Name
Budget
Club
President
1
Sponsor
1
SSN
Name
Professor
Email
E-ID
hire_date

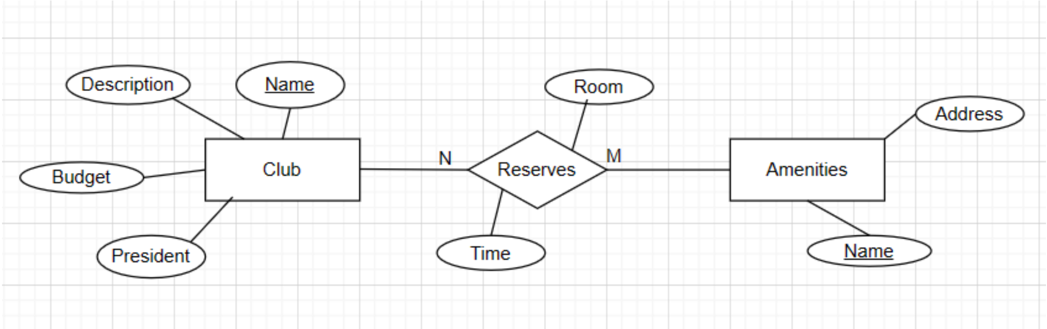
- 
section_num
section_id
Section
M
Attend
N
S_ID
name
email
birthday
Student
ssn
address

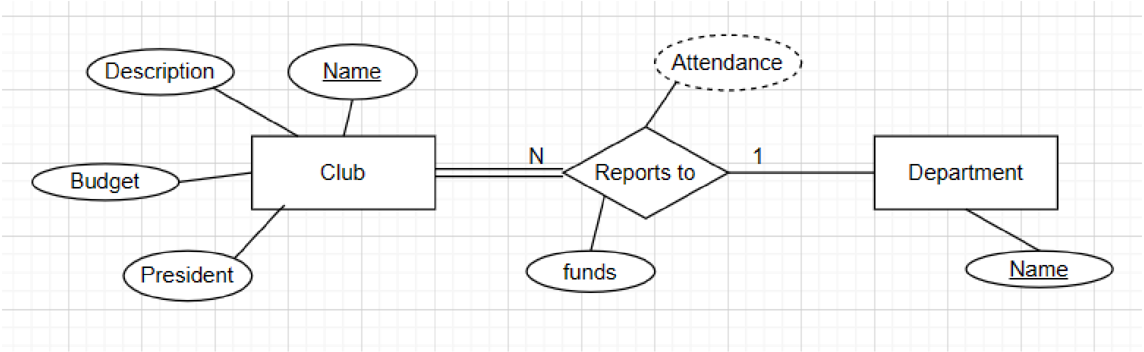
- 
Name
description
project
Lead
1
1
SSN
E-ID
name
Professor
email
hire_date

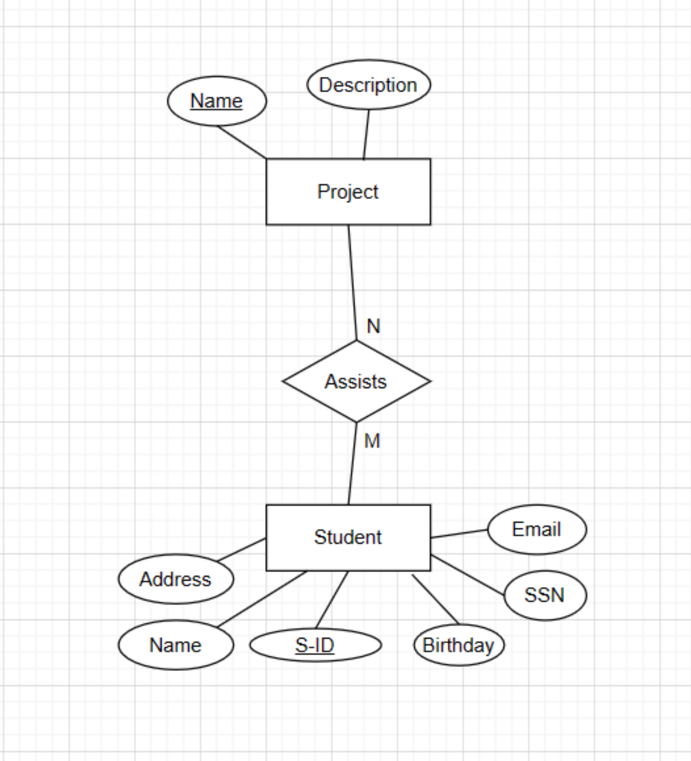
- 
Name
President
Budget
Description
Club
N
Part_of
M
Student
SSN
Name
Birthday
Email
S-ID
Address

- 
Name
President
Budget
Description
Club
N
Manages
M
Student
SSN
Name
Birthday
Email
S-ID
Address

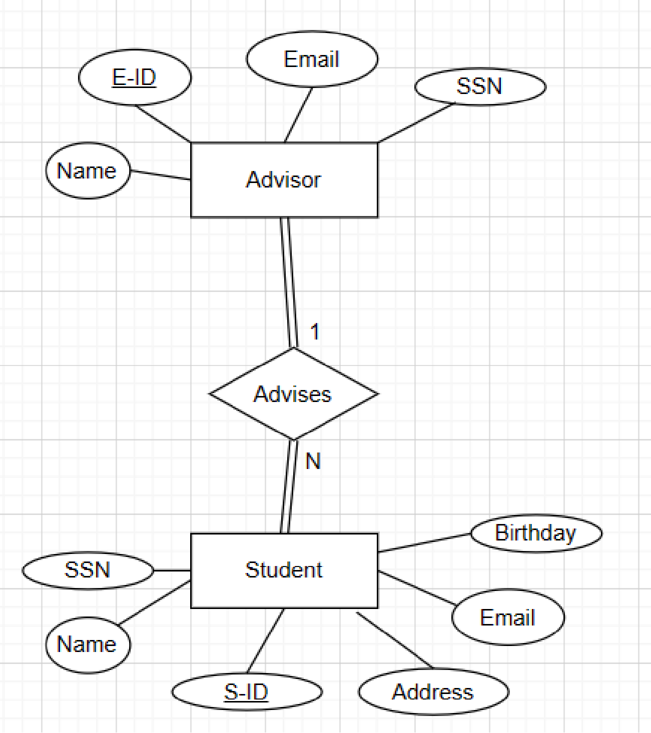
- 
E-ID
Salary
SSN
Name
Staff
N
Works_As
M
Student
Birthday
SSN
Email
Name
S-ID
Address

- 
Description
Name
Room
Address
Budget
Club
N
Reserves
M
Amenities
President
Time
Name

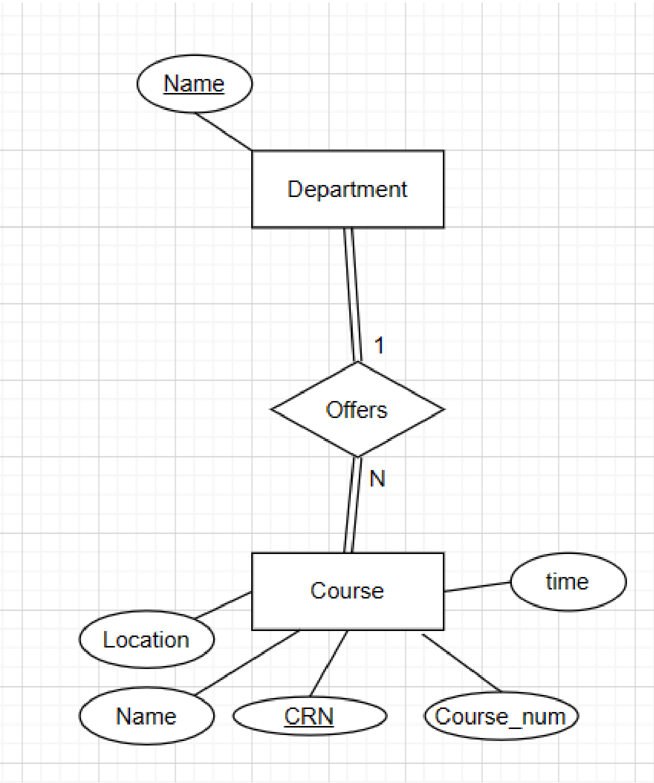
- 
Description
Name
Attendance
Budget
Club
N
Reports to
1
Department
President
funds
Name

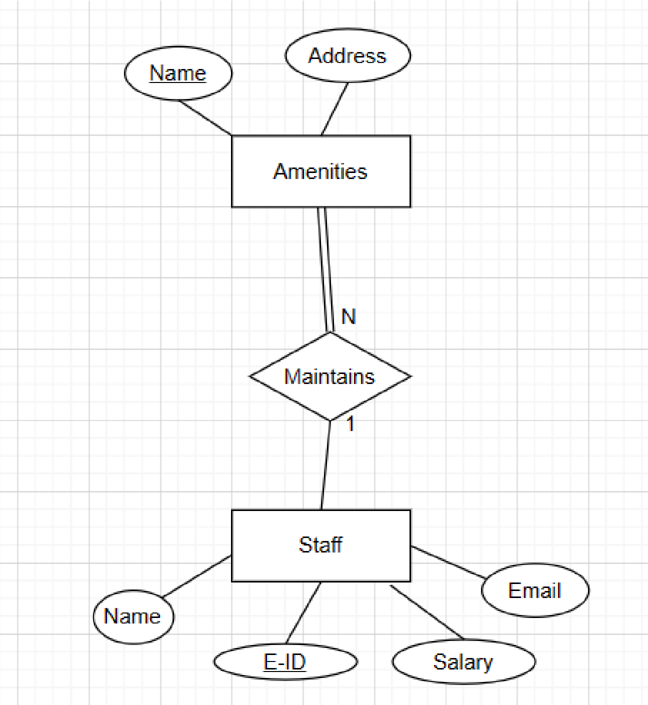

Name
Description
Project
N
Assists
M
Student
Email
Address
SSN
Name
S-ID
Birthday

-


E-ID
Email
SSN
Name
Advisor
1
Advises
N
Birthday
SSN
Student
Email
Name
S-ID
Address

-

- 
Name
Department
1
Offers
N
Course
time
Location
Name
CRN
Course_num

- 
Name
Address
Amenities
N
Maintains
1
Staff
Email
Name
E-ID
Salary

Relational Model

- Course (CRN, Name, Course_Num, Time, Location, dName)
  For this relation, we have all the attributes for the course entity and since we have a 1-to-N relationship with the department entity, instead of making a separate table for that relationship, we added dName as a foreign key to this table.
  F = {CRN -> Name, Course_num, Time, Location), this relation is in 3NF
- Section (Section_ID, Section_num, CRN)
  For this relation, we have all the attributes in the section entity, and since we have total participation with the course and each course is divided into different sections with a section number, we also introduced the CRN from the course table as a foreign key. We also have a unique value for section_ID that act as the primary key and identifies a row of record, this relation is in 3NF
- Project (Name, Description, Leading_EID)
  For this relation, we have all the attributes from the project entity and since we have a relationship that each project has to have a leading professor, we introduced leading_EID as a foreign key to the table.
  F = {Name -> Description}, this relation is in 3NF
- Attend(s-ID, Section-ID)
  We have a M-to-N relationship with students attending sections, so we created a table with sID and Section-ID, the foreign key from the two tables acting as the primary key for this relation. (this relation is in 3NF)
- Professor(E-ID, SSN, Name, Email, Hire_date)
  This relation includes all the attributes from the professor entity with E-ID as the primary key.
  F = {E-ID -> Name, Email, Hire_date}, this relation is in 3NF
- Teach(E-ID, CRN)
  We have a many to many relationship where a professor can teach many courses and a course can be taught by many professors, so we create a table for this relationship with E-ID, CRN as foreign key from the two relations and together act as the primary key of the table.
- Manages(cName, s-ID)
  We have a many to many relationship between the student entity and club entity where a student can manage many clubs and a club can be managed by many students. This relation has cName and S-ID as foreign keys and together they act as the primary key of this relation.

- Reserve(cName, aName, Room, Time)
  We have a relationship in our ER design where a club can reserve an amenity for activities, for each reservation record, we also store the room and time information. The club name and aName are the foreign keys of this table.
- Work_as(s-ID, e-ID)
  In our database design, some students may work as staff members, so we created a table with s-id and e-id as foreign keys from the student relation and the staff relation.
- Advisor(E-ID, SSN, Name, Email)
  We have the advisor entity mapped to the relational schema with E-ID as primary key and the table also includes all the attributes from the advisor entity from our ER design.
  F = {E-ID -> Name, Email}, this relation is in 3NF
- Student(s-ID, Name, Birthday, Email, SSN, Address, A-ID)
  This relation has all the attributes from the student entity from our ER design and we have a relationship between a student and an advisor where every student has to have one assigned advisor. Instead of making a new table for this relationship, we simply introduced the advisor's E-ID as a foreign key to this table.
  F = {S-ID -> Name, Birthday, Email, Address}, this relation is in 3NF
- TA (CRN, Section__ID, Office_hour, Location)
  We have a relationship in our ER diagram between the student entity and section entity, we introduced CRN and section-id as foreign keys for this relationship and together act as primary key for this table. We also included the attributes associated with this relationship in this table such as office_hour and location.
- Staff(E-ID, Salary, Name, Email)
  This relation stores all the attributes from the staff entity in our ER design with E-ID as the primary key of the table.
  F = {E-ID -> salary, name, email}
- Club(Name, Description, Budget, President, Sponsor_E-ID, dName, Funds)
  This relation stores all the attributes from the club entity from our ER design and since we have a relationship where each club has to have one sponsor and be associated with a department with a fixed amount of fund, we added the sponsor's E-ID as well as dName as foreign keys to

this table as well as funds that's associated with this relationship.
F = {Name -> description, budget, president}

- Part_of(cName, s-ID)
  We have a relationship in our ER design between the club entity and the student entity where a student can be a part of many clubs and a club can have many students. We used cName and s-ID as foreign keys for the two entities and the two together act as the primary key of this table.
- Amenities(Name, Address, E-ID)
  We have the entity amenities in our ER design and we have included all the attributes from the entity into our relation and since we have a relationship between amenity and staff where each amenity has to have one staff member doing the maintenance. In this case, we have introduced the E-ID of the maintenance staff to the table instead of making a new relation.

Data Dictionary

Table	Attribute	Description	Data Type	Constraint	Key	FK Ref. Table
Course	CRN	Course Reference Number	INT	NOT NULL, UNIQUE	PK	
	Name	Name of the course	VARCHAR(255)	NOT NULL		
	cNum	Course Number (i.e MATH1001)	VARCHAR(255)	NOT NULL		
	Time	Time for the course	VARCHAR(255)	NOT NULL		
	Location	Location of the course	VARCHAR(255)	NOT NULL		
	dName	Department course belongs to	VARCHAR(255)	NOT NULL	FK	Department
Section	sectionID	ID for the section	INT	NOT NULL, UNIQUE	PK	
	CRN	Course CRN	INT	NOT NULL, UNIQUE	PK	
	sNum	Section number for the course	INT	NOT NULL		
Project	Name	Project Name	VARCHAR(255)	NOT NULL, UNIQUE	PK	
	Description	Project Description	VARCHAR(255)	NOT NULL		
	pLead	Leading professor eID of the project	INT	NOT NULL	FK	Professor
Attends	sID	Student ID number	INT	NOT NULL, UNIQUE	PK, FK	Student
	sectionID	Section ID	INT	NOT NULL, UNIQUE	PK, FK	Section
Professor	eID	Employee ID number	INT	NOT NULL, UNIQUE	PK	
	SSN	Prof. SSN	INT	NOT NULL		
	Name	Prof. Name	VARCHAR(255)	NOT NULL		
	Email	Prof. Email	VARCHAR(255)	NOT NULL		

	HireDate	Prof. Hire date	DATE	NOT NULL		
	dName	Department Prof. works for	VARCHAR(255)	NOT NULL	FK	Department
Student	sID	Student ID number	INT	NOT NULL, UNIQUE	PK	
	Name	Student Name	VARCHAR(255)	NOT NULL		
	Bday	Student Birthday	DATE	NOT NULL		
	Email	Student Email	VARCHAR(255)	NOT NULL		
	SSN	Student SSN	INT	NOT NULL		
	Address	Student Address	VARCHAR(255)	NOT NULL		
	advisorID	Advisor’s eID	INT	NOT NULL	FK	Advisor
Department	Name	Department Name	VARCHAR(255)	NOT NULL, UNIQUE	PK	
Advisor	eID	Employee ID Number	INT	NOT NULL, UNIQUE	PK	
	SSN	Advisor SSN	INT	NOT NULL		
	Name	Advisor Name	VARCHAR(255)	NOT NULL		
	Email	Advisor Email	VARCHAR(255)	NOT NULL		
Club	Name	Club Name	VARCHAR(255)	NOT NULL, UNIQUE	PK	
	Description	Club Description	VARCHAR(255)	NOT NULL		
	Budget	Club Budget	INT	NOT NULL		
	President	Club President sID	INT	NOT NULL	FK	Student
	Sponsor	Club Sponsoring Professor eID	INT	NOT NULL	FK	Professor
	dName	Department Name that club reports to	VARCHAR(255)	NOT NULL	FK	Department
	Funds	Club Funds	INT	NOT NULL		
Amenity	Name	Amenity Name	VARCHAR(255)	NOT NULL, UNIQUE	PK	
	Address	Amenity Address	VARCHAR(255)	NOT NULL		

	M_EID	Employee ID number of the staff that maintains the amenity	INT	NOT NULL		
Staff	eID	Employee ID number	INT	NOT NULL, UNIQUE	PK	
	Salary	Staff salary	INT	NOT NULL		
	Name	Staff name	VARCHAR(255)	NOT NULL		
	Email	Staff email	VARCHAR(255)	NOT NULL		
TA	sID	Student ID number	INT	NOT NULL, UNIQUE	PK, FK	Student
	sectionID	Section ID the student TAs for	INT	NOT NULL, UNIQUE	PK, FK	Section
	Office_hour	TA's office hour	VARCHAR(255)	NOT NULL		
	Location	TA's office location	VARCHAR(255)	NOT NULL		
Teaches	eID	Prof. eID	INT	NOT NULL, UNIQUE	PK, FK	Professor
	CRN	CRN of course Prof. teaches	INT	NOT NULL, UNIQUE	PK, FK	Course
Assist	sID	Assistant's sID (student ID number)	INT	NOT NULL, UNIQUE	PK, FK	Student
	pName	Project they assist on	VARCHAR(255)	NOT NULL, UNIQUE	PK, FK	Project
Manages	cName	Name of the club	VARCHAR(255)	NOT NULL, UNIQUE	PK, FK	Club
	sID	sIDs of students that partake in management	INT	NOT NULL, UNIQUE	PK, FK	Student
Part_of	cName	Name of the club	VARCHAR(255)	NOT NULL, UNIQUE	PK, FK	Club
	sID	sIDs of students that are part of the club	INT	NOT NULL, UNIQUE	PK, FK	Student
Student_Staff	sID	Student ID number	INT	NOT NULL, UNIQUE	PK, FK	Student

	eID	Employee ID number of the student worker	INT	NOT NULL	PK, FK	Staff
Reserve	clubName	Name of the club	VARCHAR(255)	NOT NULL, UNIQUE	PK, FK	Club
	aName	The name of the amenity they’re reserving the space in	VARCHAR(255)	NOT NULL, UNIQUE	PK, FK	Amenity
	Room	The room of the amenity they’re reserving	VARCHAR(255)	NOT NULL, UNIQUE	PK	
	reserveTime	Time (date??) the club is reserving the room for	VARCHAR(255)	NOT NULL, UNIQUE	PK	

Summary

Designing and implementing this database system showed us how databases are made and what decisions go into making them.

Decisions included: What should be separate entities? What should be left as attributes? Which relationships should be represented as tables or attributes?

Constraints like UNIQUE and PRIMARY KEY help reduce redundancy and duplicated data.

Relations representing relationships were represented as unique pairs of foreign keys with an ID number for use as the primary key for MySQL.

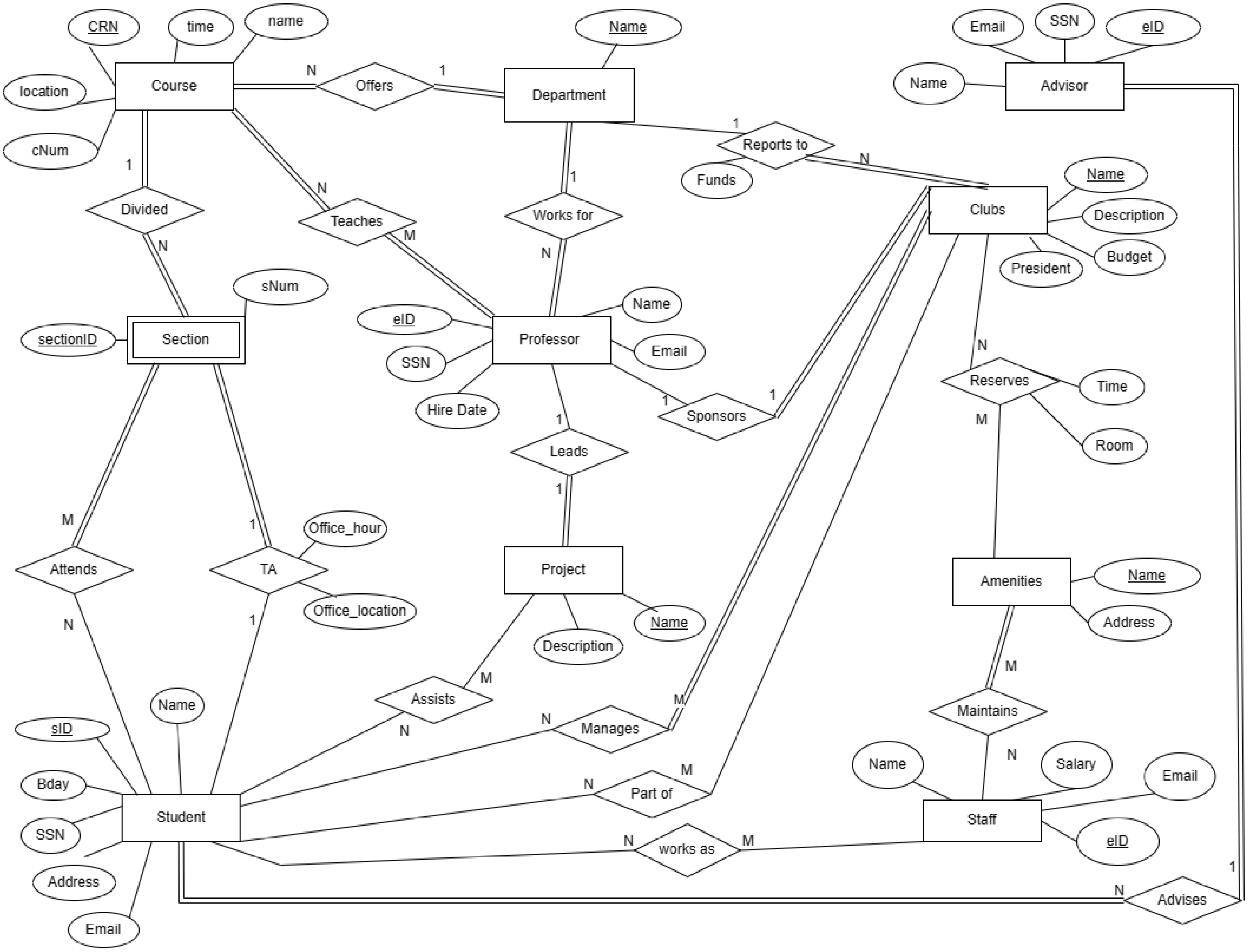
Reducing it to 3NF was easy due to the simple nature of the database.

Demo

https://youtu.be/FxEvurPxGqM

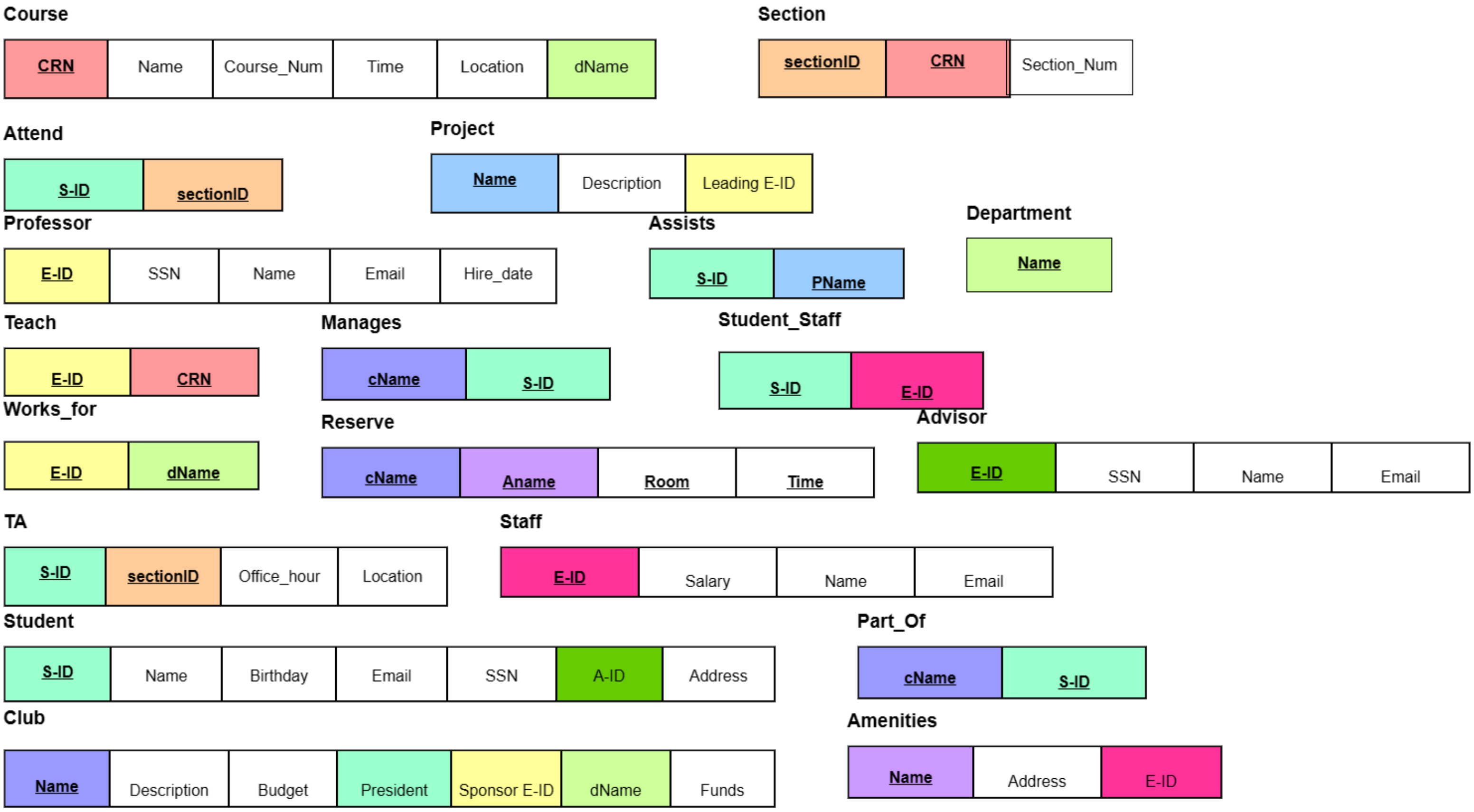
Appendix (ER Model, Relational Model, SQL code, Screenshots)

ER Model



https://drive.google.com/file/d/1sTaTZDwOZWYkeZVUT39iCvzJjxx081n9/view?usp=share_link

Relational Model



Course

CRN	Name	Course_Num	Time	Location	dName

Section

sectionID	CRN	Section_Num

Attend

S-ID	sectionID

Project

Name	Description	Leading E-ID

Professor

E-ID	SSN	Name	Email	Hire_date

Assists

S-ID	PName

Department

Name

Teach

E-ID	CRN

Manages

cName	S-ID

Student_Staff

S-ID	E-ID

Works_for

E-ID	dName

Reserve

cName	Aname	Room	Time

Advisor

E-ID	SSN	Name	Email

TA

S-ID	sectionID	Office_hour	Location

Staff

E-ID	Salary	Name	Email

Student

S-ID	Name	Birthday	Email	SSN	A-ID	Address

Part_Of

cName	S-ID

Club

Name	Description	Budget	President	Sponsor E-ID	dName	Funds

Amenities

Name	Address	E-ID

https://drive.google.com/file/d/1XrqPkx9yfknD0OfjmugxIIxHZvswsbzb/view?usp=sharing

Code:

create.sql

```
CREATE SCHEMA db_project3;
USE db_project3;


-- Entities
CREATE TABLE Course (
     CRN int NOT NULL,
    Name varchar(255) NOT NULL,
    cNum varchar(255) NOT NULL,
    Time varchar(255) NOT NULL,
    Location varchar(255) NOT NULL,
    dName varchar(255) NOT NULL,
    PRIMARY KEY (CRN)
);

CREATE TABLE Section (
     sNum int NOT NULL,
    sectionID int NOT NULL AUTO_INCREMENT,
    CRN int NOT NULL,
    PRIMARY KEY (sectionID, CRN)
);

CREATE TABLE Professor (
     eID int NOT NULL,
    SSN int NOT NULL,
    Name varchar(255) NOT NULL,
    Email varchar(255) NOT NULL,
    HireDate DATE NOT NULL,
    dName varchar(255) NOT NULL,
    PRIMARY KEY (eID)
);

CREATE TABLE Student (
     sID int NOT NULL,
    Name varchar(255) NOT NULL,
    Bday DATE NOT NULL,
    SSN int NOT NULL,
    Address varchar(255) NOT NULL,
    Email varchar(255) NOT NULL,
    advisorID int NOT NULL,

    PRIMARY KEY (sID)
);

CREATE TABLE Department (
     Name varchar(255) NOT NULL,
    PRIMARY KEY (Name)
);

CREATE TABLE Advisor (
     eID int NOT NULL,
    SSN int NOT NULL,
    Email varchar(255) NOT NULL,
    Name varchar(255) NOT NULL,
    PRIMARY KEY (eID)
);

CREATE TABLE Project (
     Name varchar(255) NOT NULL,
    Description varchar(255) NOT NULL,
    pLead int NOT NULL,
    PRIMARY KEY (Name)
);

CREATE TABLE Club (
     Name varchar(255) NOT NULL,
    Description varchar(255) NOT NULL,
    Budget int NOT NULL,
    President int NOT NULL,
    Sponsor int NOT NULL,
    Funds int NOT NULL,
    dName varchar(255) NOT NULL,
    PRIMARY KEY (Name)
);

CREATE TABLE Amenity (
     Name varchar(255) NOT NULL,
    Address varchar(255) NOT NULL,
    M_EID int NOT NULL,
    PRIMARY KEY (Name)
);

CREATE TABLE Staff (
     eID int NOT NULL,
    Salary int NOT NULL,
    Name varchar(255) NOT NULL,
    Email varchar(255) NOT NULL,
    PRIMARY KEY (eID)
);

CREATE TABLE TA (
     sID int NOT NULL,
    sectionID int NOT NULL,
    Office_Hours varchar(255) NOT NULL,
    Location varchar(255) NOT NULL,
    PRIMARY KEY (sID, sectionID)
);


-- Relations
-- CREATE TABLE Offers (
--   dName varchar(255) NOT NULL,
--     CRN int NOT NULL,
--     rID int NOT NULL AUTO_INCREMENT,
--     PRIMARY KEY (rID)
-- );

-- CREATE TABLE Reports_To (
--   dName varchar(255) NOT NULL,
--     clubName varchar(255) NOT NULL,
--   rID int NOT NULL AUTO_INCREMENT,
--   PRIMARY KEY (rID)
-- );

CREATE TABLE Attends (
     sID int NOT NULL,
    sectionID int NOT NULL,
    -- rID int NOT NULL AUTO_INCREMENT,
    PRIMARY KEY (sID, sectionID)
```

```
);

-- CREATE TABLE TA (
--   sID int NOT NULL,
--     sectionID int NOT NULL,
--     rID int NOT NULL AUTO_INCREMENT,
--     PRIMARY KEY (sID, sectionID, rID),
--     FOREIGN KEY (sectionID) REFERENCES Section(sectionID),
--     FOREIGN KEY (sID) REFERENCES Student(sID)
-- );

CREATE TABLE Teaches (
     eID int NOT NULL,
    -- rID int NOT NULL AUTO_INCREMENT,
    CRN int NOT NULL,
    PRIMARY KEY (eID, CRN)
);

CREATE TABLE Works_For (
     dName varchar(255) NOT NULL,
    eID int NOT NULL,
     -- rID int NOT NULL AUTO_INCREMENT,
     PRIMARY KEY (dName, eID)
);

-- CREATE TABLE Sponsor (
--   clubName varchar(255) NOT NULL,
--     eID int NOT NULL,
--   rID int NOT NULL AUTO_INCREMENT,
--   PRIMARY KEY (rID)
-- );

CREATE TABLE Assist (
     pName varchar(255) NOT NULL,
    sID int NOT NULL,
     -- rID int NOT NULL AUTO_INCREMENT,
     PRIMARY KEY (pName, sID)
);

CREATE TABLE Manages (
     cName varchar(255) NOT NULL,
    sID int NOT NULL,
     -- rID int NOT NULL AUTO_INCREMENT,
     PRIMARY KEY (cName, sID)
);

CREATE TABLE Part_Of (
     cName varchar(255) NOT NULL,
    sID int NOT NULL,
     -- rID int NOT NULL AUTO_INCREMENT,
     PRIMARY KEY (cName, sID)
);

CREATE TABLE Student_Staff (
     eID int NOT NULL,
    sID int NOT NULL,
     -- rID int NOT NULL AUTO_INCREMENT,
     PRIMARY KEY (eID, sID)
);

CREATE TABLE Advise (
     sID int NOT NULL,
    eID int NOT NULL,
     -- rID int NOT NULL AUTO_INCREMENT,
     PRIMARY KEY (sID, eID)
);


-- CREATE TABLE Maintain (
--   aName varchar(255) NOT NULL,
--     eID int NOT NULL,
--   rID int NOT NULL AUTO_INCREMENT,
--   PRIMARY KEY (aName, eID, rID),
--     FOREIGN KEY (aName) REFERENCES Amenity(Name),
--     FOREIGN KEY (eID) REFERENCES Staff(eID)
-- );

CREATE TABLE Reserve (
     clubName varchar(255) NOT NULL,
    aName varchar(255) NOT NULL,
    room varchar(255) NOT NULL,
    reserveTime varchar(255) NOT NULL,
     -- rID int NOT NULL AUTO_INCREMENT,
     PRIMARY KEY (clubName, aName, room, reserve)
);

-- CREATE TABLE Project_Lead (
--   pName varchar(255) NOT NULL,
--     eID int NOT NULL,
--   rID int NOT NULL AUTO_INCREMENT,
--   PRIMARY KEY (rID)
-- );

-- Add constraints

-- One TA per section, TA's can only be assigned to one section
-- ALTER TABLE Section
-- ADD CONSTRAINT UNIQUE (TA); -- Adding the same TA twice
requires a duplicate TA id
ALTER TABLE Section
ADD CONSTRAINT UNIQUE (sNum, CRN); -- Adding a second TA will
require duplicates of sNum and CRN
-- ALTER TABLE Attends
-- ADD CONSTRAINT UNIQUE (sID, sectionID); -- Registering the
same student to the same section is not allowed

-- ALTER TABLE Teaches
-- ADD CONSTRAINT UNIQUE (eID, CRN); -- Assigning the same
teacher to the same course is not allowed

-- -- No duplicate relations
-- ALTER TABLE Assist
-- ADD CONSTRAINT UNIQUE (pName, sID);

-- ALTER TABLE Manages
-- ADD CONSTRAINT UNIQUE (cName, sID);

-- ALTER TABLE Part_Of
-- ADD CONSTRAINT UNIQUE (cName, sID);

-- ALTER TABLE Student_Staff
-- ADD CONSTRAINT UNIQUE (sID, eID);
-- No making the same reservation twice
-- ALTER TABLE Reserve
-- ADD CONSTRAINT UNIQUE (aName, room, reserveTime);
```

insert.sql

```
INSERT INTO Department (Name) VALUES
     ('Mathematics'),
     ('Computer Science');

INSERT INTO Professor (eID, SSN, Name, hireDate, Email, dName)
VALUES
     ('100001', '2480', 'Lisa', '2002-06-24', 'lisa',
'Mathematics'),
     ('100002', '2341', 'John', '2000-09-17', 'john',
'Mathematics'),
     ('100003', '2567', 'Mike', '1999-05-08', 'mike',
'Mathematics'),
     ('100004', '3548', 'Will', '2003-02-09', 'will',
'Mathematics'),
     ('100005', '5678', 'Bill', '1997-12-29', 'bill',
'Mathematics'),
     ('100006', '9987', 'Charlie', '2001-01-01', 'charlie',
'Mathematics'),
     ('100007', '1245', 'Tommy', '2002-03-20', 'tommy',
'Mathematics'),
     ('100008', '3478', 'Amy', '2000-11-12', 'amy', 'Computer
Science'),
     ('100009', '5691', 'Erin', '2003-07-16', 'erin', 'Computer
Science'),
     ('100010', '8721', 'Taylor', '1996-09-30', 'taylor',
'Computer Science'),
     ('100011', '1123', 'May', '2001-04-05', 'may', 'Computer
Science'),
     ('100012', '5466', 'Sara', '1999-11-01', 'sara', 'Computer
Science'),
     ('100013', '4355', 'Julie', '2003-06-15', 'julie',
'Computer Science'),
     ('100014', '8742', 'Ben', '2000-08-28', 'ben', 'Computer
Science'),
     ('100015', '5880', 'Bill', '1995-05-23', 'bill', 'Computer
Science');

INSERT INTO Student (sID, Name, bDay, email, SSN, advisorID,
Address) VALUES
     ('2000001', 'Matthew', '2002-06-24', 'matthew', '2468',
'100020', '10 15th Street NW'),
     ('2000002', 'Victoria', '2000-09-17', 'victoria', '9763',
'100020', '22 14th Street SE'),
     ('2000003', 'Owen', '1999-05-08', 'owen', '5839', '100020',
'11 15th Street NW'),
     ('2000004', 'Zoey', '2001-08-14', 'zoey', '3057', '100020',
'23 14th Street SE'),
     ('2000005', 'Joseph', '2003-02-09', 'joseph', '1289',
'100020', '12 15th Street NW'),
     ('2000006', 'Penelope', '1997-12-29', 'penelope', '5093',
'100020', '24 14th Street SE'),
     ('2000007', 'Henry', '2001-01-01', 'henry', '7126',
'100020', '13 15th Street NW'),
     ('2000008', 'Lila', '2002-03-20', 'lila', '7642', '100020',
'25 14th Street SE'),
     ('2000009', 'Leo', '2000-11-12', 'leo', '6321', '100021',
'14 15th Street NW'),
     ('2000010', 'Levi', '2003-07-16', 'levi', '8954', '100021',
'26 14th Street SE'),
     ('2000011', 'Eleanor', '1996-09-30', 'eleanor', '2716',
'100021', '15 15th Street NW'),
     ('2000012', 'Wyatt', '2001-04-05', 'wyatt', '4871',
'100021', '27 14th Street SE'),
     ('2000013', 'Stella', '1999-11-01', 'stella', '3406',
'100021', '16 15th Street NW'),
     ('2000014', 'Caleb', '2003-06-15', 'caleb', '7154',
'100021', '28 14th Street SE'),
     ('2000015', 'Maya', '2000-08-28', 'maya', '9236', '100021',
'17 15th Street NW'),
     ('2000016', 'Julian', '1995-05-23', 'julian', '4137',
'100022', '29 14th Street SE'),
     ('2000017', 'Hannah', '2003-03-06', 'hannah', '7512',
'100022', '18 15th Street NW'),
     ('2000018', 'Ryan', '2002-12-31', 'ryan', '6104', '100022',
'30 14th Street SE'),
     ('2000019', 'Addison', '1997-01-20', 'addison', '4389',
'100022', '19 15th Street NW'),
     ('2000020', 'Isaac', '2000-07-11', 'isaac', '1829',
'100022', '31 14th Street SE'),
     ('2000021', 'Levi', '2001-10-17', 'levi', '7593', '100022',
'20 15th Street NW'),
     ('2000022', 'Eleanor', '1999-02-14', 'eleanor', '5413',
'100022', '32 14th Street SE'),
     ('2000023', 'Wyatt', '2002-04-12', 'wyatt', '3691',
'100022', '21 15th Street NW'),
     ('2000024', 'Stella', '1998-11-05', 'stella', '8207',
'100022', '33 14th Street SE'),
     ('2000025', 'Caleb', '1995-09-07', 'caleb', '6231',
'100022', '22 15th Street NW'),
     ('2000026', 'Maya', '2000-05-10', 'maya', '4019', '100023',
'34 14th Street SE'),
     ('2000027', 'Julian', '1995-06-08', 'julian', '2975',
'100023', '23 15th Street NW'),
     ('2000028', 'Hannah', '2000-08-18', 'hannah', '8705',
'100023', '35 14th Street SE'),
     ('2000029', 'Ryan', '1997-01-25', 'ryan', '9305', '100023',
'24 15th Street NW'),
     ('2000030', 'Addison', '2002-04-28', 'addison', '1763',
'100023', '36 14th Street SE'),
     ('2000031', 'Isaac', '2001-07-02', 'isaac', '5820',
'100023', '25 15th Street NW'),
     ('2000032', 'Alexander', '1999-11-23', 'alexander', '6247',
'100023', '37 14th Street SE'),
     ('2000033', 'Grace', '1997-02-27', 'grace', '8045',
'100023', '26 15th Street NW'),
     ('2000034', 'Samuel', '2000-12-22', 'samuel', '3690',
'100023', '38 14th Street SE'),
     ('2000035', 'Chloe', '1999-06-19', 'chloe', '2598',
'100023', '27 15th Street NW'),
     ('2000036', 'Daniel', '2001-10-07', 'daniel', '1092',
'100023', '39 14th Street SE'),
     ('2000037', 'Scarlett', '2002-03-15', 'scarlett', '9760',
'100023', '28 15th Street NW'),
     ('2000038', 'Matthew', '2002-01-12', 'matthew', '5831',
'100024', '40 14th Street SE'),
     ('2000039', 'Victoria', '2001-04-16', 'victoria', '4907',
'100024', '29 15th Street NW'),
     ('2000040', 'Owen', '1999-09-22', 'owen', '7653', '100024',
'41 14th Street SE'),
     ('2000041', 'Zoey', '1996-02-03', 'zoey', '6213', '100024',
'30 15th Street NW'),
     ('2000042', 'Joseph', '2002-11-14', 'joseph', '8453',
'100024', '42 14th Street SE'),
     ('2000043', 'Penelope', '2000-08-11', 'penelope', '3947',
'100024', '31 15th Street NW'),
     ('2000044', 'Henry', '2003-05-01', 'henry', '9615',
'100024', '43 14th Street SE'),
     ('2000045', 'Lila', '1996-07-27', 'lila', '5301', '100024',
'32 15th Street NW'),
     ('2000046', 'Leo', '1999-04-24', 'leo', '2675', '100024',
'44 14th Street SE'),
     ('2000047', 'Audrey', '2001-12-08', 'audrey', '7492',
'100024', '33 15th Street NW'),
     ('2000048', 'Gabriel', '2000-03-12', 'gabriel', '4806',
'100025', '45 14th Street SE'),
     ('2000049', 'Lily', '1998-09-21', 'lily', '6492', '100025',
'34 15th Street NW'),
     ('2000050', 'Levi', '2001-06-17', 'levi', '5304', '100025',
'46 14th Street SE'),
     ('2000051', 'Eleanor', '1997-10-25', 'eleanor', '6531',
'100025', '35 15th Street NW'),
     ('2000052', 'Wyatt', '2000-01-09', 'wyatt', '7592',
'100025', '47 14th Street SE'),
     ('2000053', 'Stella', '2002-12-05', 'stella', '4958',
'100025', '36 15th Street NW'),
     ('2000054', 'Caleb', '1995-05-26', 'caleb', '1234',
'100025', '48 14th Street SE'),
     ('2000055', 'Maya', '1997-03-18', 'maya', '8743', '100025',
'37 15th Street NW'),
     ('2000056', 'Julian', '2000-02-10', 'julian', '9637',
'100025', '49 14th Street SE'),
     ('2000057', 'Hannah', '1995-09-11', 'hannah', '5826',
'100025', '38 15th Street NW'),
     ('2000058', 'Ryan', '1998-08-08', 'ryan', '6890', '100025',
'50 14th Street SE'),
     ('2000059', 'Addison', '1999-11-08', 'addison', '2468',
'100026', '39 15th Street NW'),
     ('2000060', 'Isaac', '1995-07-08', 'isaac', '5217',
'100026', '51 14th Street SE'),
     ('2000061', 'Ava', '2002-01-22', 'ava', '7845', '100026',
'40 15th Street NW'),
     ('2000062', 'Adam', '1999-06-09', 'adam', '9672', '100026',
'52 14th Street SE'),
     ('2000063', 'Caroline', '2001-10-15', 'caroline', '4316',
'100026', '41 15th Street NW'),
     ('2000064', 'David', '1998-05-11', 'david', '2957',
'100026', '53 14th Street SE'),
     ('2000065', 'Leah', '1997-12-26', 'leah', '6785', '100026',
'42 15th Street NW'),
     ('2000066', 'Nathan', '2000-03-24', 'nathan', '3158',
'100026', '54 14th Street SE'),
     ('2000067', 'Ruby', '1995-07-29', 'ruby', '8764', '100026',
'43 15th Street NW'),
     ('2000068', 'Isaac', '2001-11-16', 'isaac', '4391',
'100027', '55 14th Street SE'),
     ('2000069', 'Gabriella', '2002-06-26', 'gabriella', '5876',
'100027', '44 15th Street NW'),
     ('2000070', 'Tyler', '2000-09-15', 'tyler', '9254',
'100027', '56 14th Street SE'),
     ('2000071', 'Claire', '1995-08-09', 'claire', '3678',
'100027', '45 15th Street NW'),
     ('2000072', 'Dylan', '1999-01-29', 'dylan', '7105',
'100027', '57 14th Street SE'),
     ('2000073', 'Mia', '2000-10-04', 'mia', '8526', '100027',
'46 15th Street NW'),
     ('2000074', 'Carson', '2003-05-30', 'carson', '3985',
'100027', '58 14th Street SE'),
     ('2000075', 'Layla', '2003-02-09', 'layla', '1062',
'100027', '47 15th Street NW'),
     ('2000076', 'Cooper', '1997-12-29', 'cooper', '9372',
'100027', '59 14th Street SE'),
     ('2000077', 'Sadie', '2001-01-01', 'sadie', '5893',
'100028', '48 15th Street NW'),
     ('2000078', 'Noah', '2002-03-20', 'noah', '2507', '100028',
'60 14th Street SE'),
     ('2000079', 'Autumn', '2000-11-12', 'autumn', '4931',
'100028', '49 15th Street NW'),
     ('2000080', 'Caleb', '2003-07-16', 'caleb', '7268',
'100028', '61 14th Street SE'),
     ('2000081', 'Ruby', '1996-09-30', 'ruby', '5190', '100028',
'50 15th Street NW'),
     ('2000082', 'Isaac', '2001-04-05', 'isaac', '8617',
'100028', '62 14th Street SE'),
     ('2000083', 'Gabriella', '1999-11-01', 'gabriella', '3156',
'100028', '51 15th Street NW'),
     ('2000084', 'Tyler', '2003-06-15', 'tyler', '6702',
'100028', '63 14th Street SE'),
     ('2000085', 'Claire', '2000-08-28', 'claire', '4128',
'100028', '52 15th Street NW'),
     ('2000086', 'Dylan', '1995-05-23', 'dylan', '9754',
'100028', '64 14th Street SE'),
     ('2000087', 'Mia', '2003-03-06', 'mia', '1845', '100028',
'53 15th Street NW'),
     ('2000088', 'Carson', '2002-12-31', 'carson', '5478',
'100028', '65 14th Street SE');

INSERT INTO Advisor (EID, SSN, Name, Email) VALUES
     ('100020', '1234', 'Henry', 'henry'),
     ('100021', '5678', 'Olivia', 'olivia'),
     ('100022', '9012', 'Ethan', 'ethan'),
     ('100023', '3456', 'Sophia', 'sophia'),
     ('100024', '7890', 'Liam', 'liam'),
     ('100025', '1645', 'Ava', 'ava'),
     ('100026', '4581', 'Benjamin', 'benjamin'),
     ('100027', '3598', 'Emma', 'emma'),
     ('100028', '1524', 'Noah', 'noah');

INSERT INTO Amenity (Name, Address, M_EID) VALUES
     ('Amenity 1', '10 15th Street NW', '100041'),
     ('Amenity 2', '22 14th Street SE', '100042');

INSERT INTO Assist (sID, PName) VALUES
     ('2000047', 'School course database'),
     ('2000054', 'School course database'),
     ('2000012', 'Math theory'),
     ('2000024', 'Math theory'),
     ('2000048', 'Network routing'),
     ('2000067', 'Network routing'),
     ('2000055', 'Network routing');

INSERT INTO Attends (sID, sectionID) VALUES
     ('2000001', '1'),
     ('2000002', '2'),
     ('2000003', '3'),
     ('2000004', '4'),
     ('2000005', '5'),
     ('2000006', '6'),
     ('2000007', '7'),
     ('2000008', '8'),
     ('2000009', '9'),
     ('2000010', '10'),
     ('2000011', '11'),
     ('2000012', '12'),
     ('2000013', '13'),
     ('2000014', '14'),
     ('2000015', '15'),
     ('2000016', '16'),
     ('2000017', '17'),
     ('2000018', '18'),
     ('2000019', '19'),
     ('2000020', '20'),
     ('2000021', '21'),
     ('2000022', '22'),
     ('2000023', '23'),
     ('2000024', '24'),
     ('2000025', '25'),
     ('2000026', '26'),
     ('2000027', '27'),
     ('2000028', '28'),
     ('2000029', '29'),
     ('2000030', '30'),
     ('2000031', '31'),
     ('2000032', '32'),
     ('2000033', '33'),
     ('2000034', '34'),
     ('2000035', '35'),
     ('2000036', '36'),
     ('2000037', '37'),
     ('2000038', '38'),
     ('2000039', '39'),
     ('2000040', '40'),
     ('2000041', '41'),
     ('2000042', '42'),
     ('2000043', '43'),
     ('2000044', '44'),
     ('2000045', '45'),
     ('2000046', '46'),
     ('2000047', '47'),
     ('2000048', '48'),
     ('2000049', '49'),
     ('2000050', '50'),
     ('2000051', '51'),
     ('2000052', '52'),
     ('2000053', '53'),
     ('2000054', '54'),
     ('2000055', '55'),
     ('2000056', '56'),
     ('2000057', '57'),
     ('2000058', '58'),
     ('2000059', '59'),
     ('2000060', '60'),
     ('2000061', '61'),
     ('2000062', '62'),
     ('2000063', '63'),
     ('2000064', '1'),
     ('2000065', '2'),
     ('2000066', '3'),
     ('2000067', '4'),
     ('2000068', '5'),
     ('2000069', '6'),
     ('2000070', '7'),
     ('2000071', '8'),
     ('2000072', '9'),
     ('2000073', '10'),
     ('2000074', '11'),
     ('2000075', '12'),
     ('2000076', '13'),
     ('2000077', '14'),
     ('2000078', '15'),
     ('2000079', '16'),
     ('2000080', '17'),
     ('2000081', '18'),
     ('2000082', '19'),
     ('2000083', '20'),
     ('2000084', '21'),
     ('2000085', '22'),
     ('2000086', '23'),
     ('2000087', '24'),
     ('2000088', '25');

INSERT INTO Club (Name, Description, Budget, president, sponsor,
dName, Funds) VALUES
     ('Computer Club', 'where students discuss topic in computer
science and have guest speakers from the industry', '2000',
'2000036', '100011', 'Computer Science', '100'),
     ('Math Club', 'where student pratice speed math and discuss
theoratical problems in math', '2300', '2000012', '100004',
'Mathematics', '200'),
     ('Girls Who Code', '"women who are in computer science
major, invite female guest speaker from the industry"', '1500',
'2000015', '100012', 'Computer Science', '300'),
     ('Puzzle Club', 'where students get together and solve all
kinds of puzzles', '1200', '2000055', '100006', 'Mathematics',
'400');

INSERT INTO Course (CRN, name, cNum, Time, location, dName)
VALUES
     ('1111', 'Algebra I', 'MATH1001', '8:00:00 - 10:00',
'Building 1', 'Mathematics'),
     ('1112', 'Algebra II', 'MATH1002', '8:00:00 - 10:00',
'Building 1', 'Mathematics'),
     ('1113', 'Advance Algebra', 'MATH 1003', '10:00 - 12:00',
'Building 1', 'Mathematics'),
     ('1114', 'Calculus I', 'MATH 2001', '10:00-11:45',
'Building 3', 'Mathematics'),
     ('1115', 'Calculus II', 'MATH 2002', '2:30 - 4:00',
'Building 2', 'Mathematics'),
     ('1116', 'Advance Calculus', 'MATH 2003', '2:30 - 4:00',
'Building 2', 'Mathematics'),
     ('1117', 'Linear Algebra', 'MATH 3001', '10:00 - 12:00',
'Building 2', 'Mathematics'),
     ('1118', 'Probability & Statistic', 'MATH 3002', '12:30 -
2:00', 'Building 3', 'Mathematics'),
     ('1119', 'Basic Programming I', 'CS 1001', '12:45 - 2:30',
'Building 3', 'Computer Science'),
     ('1120', 'Basic Programming II', 'CS 1002', '2: 00 - 3:30',
'Building 5', 'Computer Science'),
     ('1121', 'Advance Programming', 'CS 1003', '8:00 - 10:00',
'Building 2', 'Computer Science'),
     ('1122', 'Data Structure', 'CS 2001', '10:00 - 12:00',
'Building 4', 'Computer Science'),
     ('1123', 'System Level Programming', 'CS 2002', '12:45 -
2:30', 'Building 4', 'Computer Science'),
     ('1124', 'Algorithm', 'CS 3001', '2:30 - 4:00', 'Building
2', 'Computer Science'),
     ('1125', 'Software Engineering', 'CS 3002', '2:30 - 4:00',
'Building 3', 'Computer Science'),
     ('1126', 'Database System', 'CS 4001', '10:00 - 12:00',
'Building 5', 'Computer Science'),
     ('1127', 'Computer Network', 'CS 4002', '8:00 - 10:00',
'Building 5', 'Computer Science'),
     ('1128', 'Fundamental of Cybersecurity', 'CS 4003', '10:00
- 12:00', 'Building 2', 'Computer Science'),
     ('1129', 'Fundamental of Data Science', 'CS 4004', '12:45 -
2:30', 'Building 6', 'Computer Science'),
     ('1130', 'Network Security', 'CS 4005', '2: 00 - 3:30',
'Building 6', 'Computer Science');

INSERT INTO Manages (cName, sID) VALUES
     ('Computer Club', '2000010'),
     ('Computer Club', '2000040'),
     ('Computer Club', '2000008'),
     ('Math Club', '2000034'),
     ('Math Club', '2000001'),
     ('Girls Who Code', '2000008'),
     ('Girls Who Code', '2000045'),
     ('Puzzle Club', '2000033'),
     ('Puzzle Club', '2000003'),
     ('Puzzle Club', '2000005');

INSERT INTO Part_Of (cName, sID) VALUES
     ('Computer Club', '2000008'),
     ('Computer Club', '2000044'),
     ('Computer Club', '2000034'),
     ('Computer Club', '2000021'),
     ('Computer Club', '2000011'),
     ('Math Club', '2000032'),
     ('Math Club', '2000007'),
     ('Math Club', '2000009'),
     ('Math Club', '2000018'),
     ('Girls Who Code', '2000014'),
     ('Girls Who Code', '2000008'),
     ('Girls Who Code', '2000010'),
     ('Girls Who Code', '2000022'),
     ('Puzzle Club', '2000034'),
     ('Puzzle Club', '2000010'),
     ('Puzzle Club', '2000055'),
     ('Puzzle Club', '2000049');

INSERT INTO Project (Name, Description, pLead) VALUES
     ('School course database', 'implement a database storing
all the course information', '100011'),
     ('Network routing', 'implement router setup for the school
network', '100014'),
     ('Math theory', 'testing newly published math theory',
'100003');

INSERT INTO TA (sID, Office_Hours, Location) VALUES
     ('2000001', 'Mon 10:00-11:00', 'Building 1'),
     ('2000002', 'Thu 10:00 - 11:00', 'Building 2'),
     ('2000003', 'Mon 1:00 - 2:00', 'Building 1'),
     ('2000004', 'Fri 9:00-10:00', 'Building 3'),
     ('2000005', 'Tue:11:00-12:00', 'Building 4'),
     ('2000006', 'Mon 1:00 - 2:00', 'Building 1'),
     ('2000007', 'Mon 1:00 - 2:00', 'Building 6'),
     ('2000008', 'Wed 2:00 - 3:00', 'Building 1'),
     ('2000009', 'Tue 1:00 - 2:00', 'Building 2'),
     ('2000010', 'Wed 2:00 - 3:00', 'Building 5'),
     ('2000011', 'Tue 1:00 - 2:00', 'Building 6'),
     ('2000012', 'Mon 3:00 - 4:00', 'Building 4'),
     ('2000013', 'Fri 11:00 - 12:00', 'Building 2'),
     ('2000014', 'Thu 10:00 - 11:00', 'Building 6'),
     ('2000015', 'Tue 1:00 - 2:00', 'Building 1'),
     ('2000016', 'Fri 11:00 - 12:00', 'Building 2'),
     ('2000017', 'Fri 9:00-10:00', 'Building 3'),
     ('2000018', 'Wed 2:00 - 3:00', 'Building 2'),
     ('2000019', 'Tue 1:00 - 2:00', 'Building 3'),
     ('2000020', 'Wed 2:00 - 3:00', 'Building 4'),
     ('2000021', 'Mon 10:00-11:00', 'Building 1'),
     ('2000022', 'Tue 1:00 - 2:00', 'Building 5'),
     ('2000023', 'Wed 2:00 - 3:00', 'Building 2'),
     ('2000024', 'Mon 1:00 - 2:00', 'Building 4'),
     ('2000025', 'Tue 1:00 - 2:00', 'Building 6'),
     ('2000026', 'Tue 1:00 - 2:00', 'Building 5'),
     ('2000027', 'Mon 10:00-11:00', 'Building 6'),
     ('2000028', 'Fri 11:00 - 12:00', 'Building 4'),
     ('2000029', 'Wed 2:00 - 3:00', 'Building 6'),
     ('2000030', 'Tue 1:00 - 2:00', 'Building 1'),
     ('2000031', 'Mon 1:00 - 2:00', 'Building 2'),
     ('2000032', 'Mon 3:00 - 4:00', 'Building 3'),
     ('2000033', 'Fri 9:00-10:00', 'Building 2'),
     ('2000034', 'Mon 10:00-11:00', 'Building 3'),
     ('2000035', 'Mon 1:00 - 2:00', 'Building 4'),
     ('2000036', 'Fri 9:00-10:00', 'Building 1'),
     ('2000037', 'Fri 11:00 - 12:00', 'Building 5'),
     ('2000038', 'Tue 1:00 - 2:00', 'Building 2'),
     ('2000039', 'Fri 9:00-10:00', 'Building 4'),
     ('2000040', 'Tue 1:00 - 2:00', 'Building 6'),
     ('2000041', 'Wed 2:00 - 3:00', 'Building 5'),
     ('2000042', 'Thu 10:00 - 11:00', 'Building 6'),
     ('2000043', 'Fri 9:00-10:00', 'Building 4'),
     ('2000044', 'Mon 1:00 - 2:00', 'Building 4'),
     ('2000045', 'Thu 10:00 - 11:00', 'Building 1'),
     ('2000046', 'Tue 1:00 - 2:00', 'Building 6'),
     ('2000047', 'Wed 2:00 - 3:00', 'Building 1'),
     ('2000048', 'Thu 10:00 - 11:00', 'Building 2'),
     ('2000049', 'Fri 9:00-10:00', 'Building 5'),
     ('2000050', 'Mon 10:00-11:00', 'Building 6'),
     ('2000051', 'Mon 1:00 - 2:00', 'Building 4'),
     ('2000052', 'Fri 9:00-10:00', 'Building 2'),
     ('2000053', 'Fri 11:00 - 12:00', 'Building 6'),
     ('2000054', 'Tue 1:00 - 2:00', 'Building 1'),
     ('2000055', 'Fri 9:00-10:00', 'Building 2'),
     ('2000056', 'Tue 1:00 - 2:00', 'Building 3'),
     ('2000057', 'Wed 2:00 - 3:00', 'Building 2'),
     ('2000058', 'Thu 10:00 - 11:00', 'Building 3'),
     ('2000059', 'Fri 9:00-10:00', 'Building 4'),
     ('2000060', 'Mon 1:00 - 2:00', 'Building 1'),
     ('2000061', 'Thu 10:00 - 11:00', 'Building 2'),
     ('2000062', 'Tue 1:00 - 2:00', 'Building 3'),
     ('2000063', 'Wed 2:00 - 3:00', 'Building 4');

INSERT INTO Reserve (clubName, aName, Room, reserveTime) VALUES
     ('Computer Club', 'Amenity 1', '100', '7/7/2023'),
     -- ('Computer Club', 'Amenity 2', '100', '7/8/2023'),
     ('Math Club', 'Amenity 2', '200', '7/7/2023'),
     ('Girls Who Code', 'Amenity 1', '100', '6/7/2023'),
     -- ('Girls Who Code', 'Amenity 2', '100', '6/12/2023'),
     ('Puzzle Club', 'Amenity 1', '200', '6/5/2023');

INSERT INTO Section (sNum, CRN, TA) VALUES
     ('1', '1111', '2000001'),
     ('2', '1111', '2000002'),
     ('3', '1111', '2000003'),
     ('1', '1112', '2000004'),
     ('2', '1112', '2000005'),
     ('3', '1112', '2000006'),
     ('1', '1113', '2000007'),
     ('2', '1113', '2000008'),
     ('3', '1113', '2000009'),
     ('1', '1114', '2000010'),
     ('2', '1114', '2000011'),
     ('3', '1114', '2000012'),
     ('1', '1115', '2000013'),
     ('2', '1115', '2000014'),
     ('3', '1115', '2000015'),
     ('4', '1115', '2000016'),
     ('1', '1116', '2000017'),
     ('2', '1116', '2000018'),
     ('3', '1116', '2000019'),
     ('1', '1117', '2000020'),
     ('2', '1117', '2000021'),
     ('3', '1117', '2000022'),
     ('4', '1117', '2000023'),
     ('1', '1118', '2000024'),
     ('2', '1118', '2000025'),
     ('3', '1118', '2000026'),
     ('1', '1119', '2000027'),
     ('2', '1119', '2000028'),
     ('3', '1119', '2000029'),
     ('1', '1120', '2000030'),
     ('2', '1120', '2000031'),
     ('3', '1120', '2000032'),
     ('1', '1121', '2000033'),
     ('2', '1121', '2000034'),
     ('3', '1121', '2000035'),
     ('1', '1122', '2000036'),
     ('2', '1122', '2000037'),
     ('3', '1122', '2000038'),
     ('1', '1123', '2000039'),
     ('2', '1123', '2000040'),
     ('3', '1123', '2000041'),
     ('4', '1123', '2000042'),
     ('1', '1124', '2000043'),
     ('2', '1124', '2000044'),
     ('3', '1124', '2000045'),
     ('1', '1125', '2000046'),
     ('2', '1125', '2000047'),
     ('3', '1125', '2000048'),
     ('1', '1126', '2000049'),
     ('2', '1126', '2000050'),
     ('3', '1126', '2000051'),
     ('1', '1127', '2000052'),
     ('2', '1127', '2000053'),
     ('3', '1127', '2000054'),
     ('1', '1128', '2000055'),
     ('2', '1128', '2000056'),
     ('3', '1128', '2000057'),
     ('1', '1129', '2000058'),
     ('2', '1129', '2000059'),
     ('3', '1129', '2000060'),
     ('1', '1130', '2000061'),
     ('2', '1130', '2000062'),
     ('3', '1130', '2000063');

INSERT INTO Staff (eID, Salary, Name, Email) VALUES
     ('100030', '3456', 'Isaac', 'isaac'),
     ('100031', '526215', 'Ava', 'ava'),
     ('100032', '1561', 'Adam', 'adam'),
     ('100033', '1951', 'Caroline', 'caroline'),
     ('100034', '51951', 'David', 'david'),
     ('100035', '2592', 'Leah', 'leah'),
     ('100036', '91', 'Nathan', 'nathan'),
     ('100037', '123132', 'Ruby', 'ruby'),
     ('100038', '43576', 'Isaac', 'isaac'),
     ('100039', '5674', 'Gabriella', 'gabriella'),
     ('100040', '324635', 'Tyler', 'tyler'),
     ('100041', '7674', 'Claire', 'claire'),
     ('100042', '324466', 'Dylan', 'dylan'),
     ('100043', '3254345', 'Mia', 'mia'),
     ('100044', '32542543', 'Carson', 'carson'),
     ('100045', '65323', 'Layla', 'layla'),
     ('100046', '132879', 'Cooper', 'cooper');

INSERT INTO Teaches (eID, CRN) VALUES
     ('100001', '1111'),
     ('100003', '1111'),
     ('100002', '1112'),
     ('100003', '1112'),
     ('100003', '1113'),
     ('100005', '1114'),
     ('100003', '1114'),
     ('100004', '1115'),
     ('100005', '1116'),
     ('100006', '1117'),
     ('100007', '1118'),
     ('100008', '1119'),
     ('100009', '1120'),
     ('100008', '1121'),
     ('100010', '1121'),
     ('100011', '1122'),
     ('100013', '1122'),
     ('100012', '1123'),
     ('100013', '1124'),
     ('100014', '1125'),
     ('100009', '1126'),
     ('100001', '1127'),
     ('100015', '1128'),
     ('100012', '1129'),
     ('100009', '1130');

INSERT INTO Student_Staff (sID, eID) VALUES
     ('2000060', '100030'),
     ('2000061', '100031'),
     ('2000062', '100032'),
     ('2000063', '100033'),
     ('2000064', '100034'),
     ('2000065', '100035'),
     ('2000066', '100036'),
     ('2000067', '100037'),
     ('2000068', '100038'),
     ('2000069', '100039'),
     ('2000070', '100040'),
     ('2000071', '100041'),
     ('2000072', '100042'),
     ('2000073', '100043'),
     ('2000074', '100044'),
     ('2000075', '100045'),
     ('2000076', '100046');


-- Add foreign Keys

ALTER TABLE Course
ADD FOREIGN KEY (dName) REFERENCES Department(Name);

ALTER TABLE Section
ADD FOREIGN KEY (CRN) REFERENCES Course(CRN);
ALTER TABLE Section
ADD  FOREIGN KEY (TA) REFERENCES Student(sID);

ALTER TABLE Club
ADD (FOREIGN KEY (President) REFERENCES Student(sID),
     FOREIGN KEY (Sponsor) REFERENCES Professor(eID),
    FOREIGN KEY (dName) REFERENCES Department(Name));

ALTER TABLE Project
ADD FOREIGN KEY (pLead) REFERENCES Professor(eID);

ALTER TABLE Amenity
ADD FOREIGN KEY (M_EID) REFERENCES Staff(eID);

-- ALTER TABLE Offers
-- ADD (FOREIGN KEY (dName) REFERENCES Department(Name),
--     FOREIGN KEY (CRN) REFERENCES Course(CRN));

-- ALTER TABLE Reports_To
-- ADD (FOREIGN KEY (dName) REFERENCES Department(Name),
--     FOREIGN KEY (clubName) REFERENCES Club(Name));

ALTER TABLE Attends
ADD (FOREIGN KEY (sectionID) REFERENCES Section(sectionID),
    FOREIGN KEY (sID) REFERENCES Student(sID));

ALTER TABLE Teaches
ADD (FOREIGN KEY (eID) REFERENCES Professor(eID),
    FOREIGN KEY (CRN) REFERENCES Course(CRN));

-- ALTER TABLE Works_For
-- ADD (FOREIGN KEY (dName) REFERENCES Department(Name),
--     FOREIGN KEY (eID) REFERENCES Professor(eID));

-- ALTER TABLE Sponsor
-- ADD (FOREIGN KEY (clubName) REFERENCES Club(Name),
--     FOREIGN KEY (eID) REFERENCES Professor(eID));

ALTER TABLE Assist
ADD (FOREIGN KEY (pName) REFERENCES Project(Name),
    FOREIGN KEY (sID) REFERENCES Student(sID));

ALTER TABLE Manages
ADD (FOREIGN KEY (cName) REFERENCES Club(Name),
    FOREIGN KEY (sID) REFERENCES Student(sID));

ALTER TABLE Part_Of
```

```
ADD (FOREIGN KEY (cName) REFERENCES Club(Name),
    FOREIGN KEY (sID) REFERENCES Student(sID));

ALTER TABLE Student_Staff
ADD (FOREIGN KEY (eID) REFERENCES Staff(eID),
    FOREIGN KEY (sID) REFERENCES Student(sID));

-- ALTER TABLE Advise
-- ADD (FOREIGN KEY (sID) REFERENCES Student(sID),
--     FOREIGN KEY (eID) REFERENCES Advisor(eID));

ALTER TABLE Reserve
ADD (FOREIGN KEY (aName) REFERENCES Amenity(Name),
    FOREIGN KEY (clubName) REFERENCES Club(Name));

-- ALTER TABLE Project_Lead
-- ADD (FOREIGN KEY (pName) REFERENCES Project(Name),
--     FOREIGN KEY (eID) REFERENCES Professor(eID));
```

queries.sql

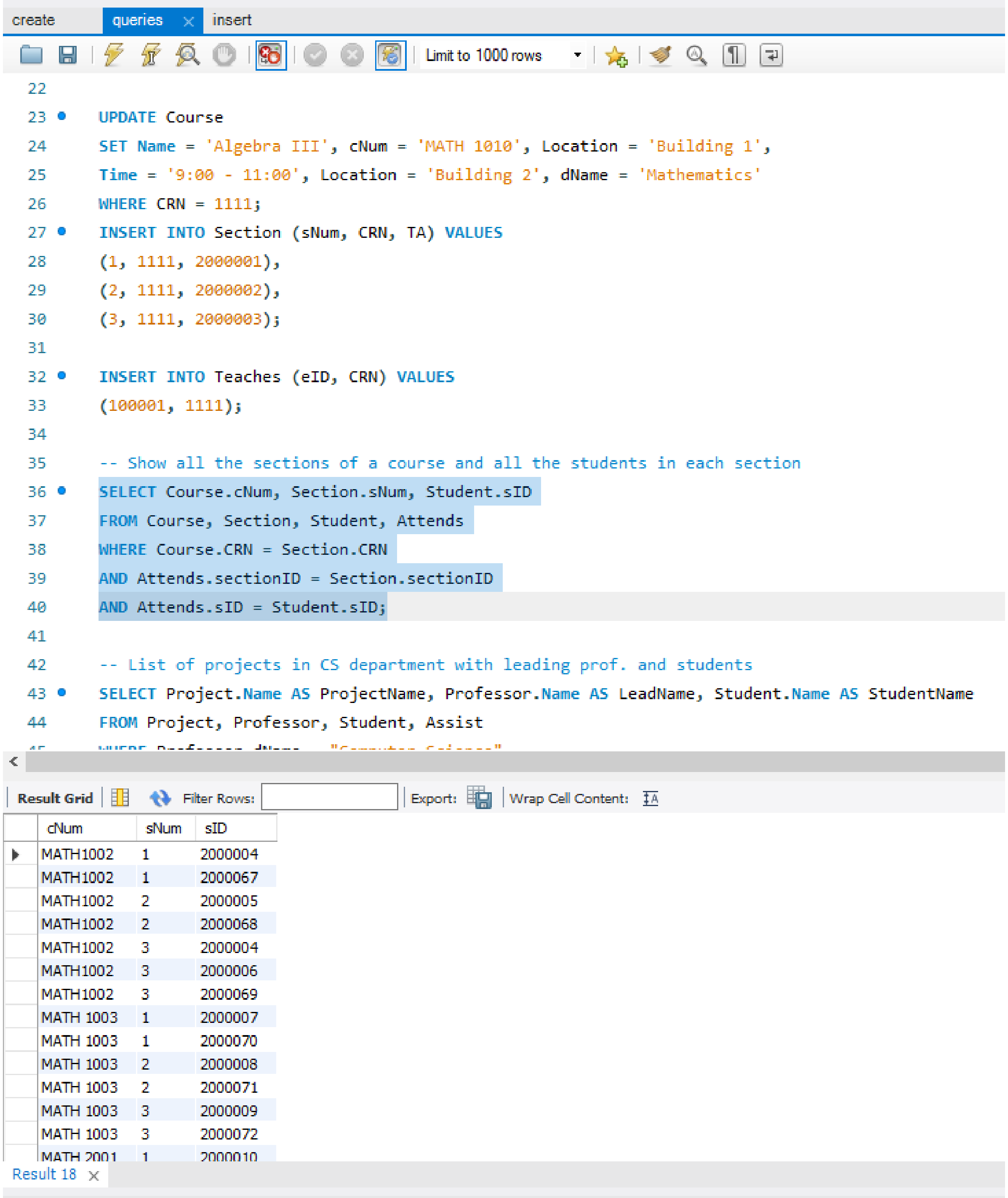
```
-- Names of professors who teach more than one class:
SELECT Professor.eID, Professor.Name, COUNT(*)
FROM Professor, Teaches
WHERE Professor.eID = Teaches.eID
GROUP BY Teaches.eID
HAVING COUNT(*) > 1;

-- Generate list of classes taught by a professor
SELECT Course.CRN, Course.cNum
FROM Course, Teaches, Professor
WHERE Course.CRN = Teaches.CRN
AND Teaches.eID = Professor.eID
AND Professor.Name = 'Lisa';

-- Replace a course with a new one
DELETE FROM Teaches
WHERE CRN = 1111;
DELETE FROM Attends
USING Attends, Section
```

```
WHERE Attends.sectionID = Section.sectionID AND Section.CRN =
1111;
DELETE FROM Section WHERE CRN = 1111;

UPDATE Course
SET Name = 'Algebra III', cNum = 'MATH 1010', Location =
'Building 1',
Time = '9:00 - 11:00', Location = 'Building 2', dName =
'Mathematics'
WHERE CRN = 1111;
INSERT INTO Section (sNum, CRN, TA) VALUES
(1, 1111, 2000001),
(2, 1111, 2000002),
(3, 1111, 2000003);

INSERT INTO Teaches (eID, CRN) VALUES
(100001, 1111);

-- Show all the sections of a course and all the students in
each section
SELECT Course.cNum, Section.sNum, Student.sID
FROM Course, Section, Student, Attends
WHERE Course.CRN = Section.CRN
AND Attends.sectionID = Section.sectionID
AND Attends.sID = Student.sID;

-- List of projects in CS department with leading prof. and
students
SELECT Project.Name AS ProjectName, Professor.Name AS LeadName,
Student.Name AS StudentName
FROM Project, Professor, Student, Assist
WHERE Professor.dName = "Computer Science"
AND Project.pLead = Professor.eID
AND Project.Name = Assist.pName
AND Assist.sID = Student.sID;

-- Clubs can reserve space in an amenity building with a stated
room and time
-- DELETE FROM Reserve
-- WHERE clubName = 'Puzzle Club' AND aName = 'Amenity 2';
INSERT INTO Reserve (clubName, aName, room, reserveTime) VALUES
```

```
('Puzzle Club', 'Amenity 2', 'bathroom', '1/1/2023');

-- Change club sponsor

UPDATE Club
SET Sponsor = 100005
WHERE Name = 'Computer Club';
-- SELECT * FROM CLUB;

-- SELECT * FROM Amenity, Staff Where Amenity.M_EID = Staff.eID;

-- Attempt to delete staff member tasked with maintenance
(should fail)
DELETE FROM Staff WHERE eID = 100041;

-- List of students under a given advisor
-- SELECT * FROM Advisor;
SELECT Student.sID, Student.Name
FROM Student, Advisor
WHERE Student.advisorID = Advisor.eID AND Advisor.Name =
"Henry";

-- Change project lead
UPDATE Project
SET pLead = 100002
WHERE Name = 'Math theory';
-- SELECT * FROM Project, Professor WHERE Project.pLead =
Professor.eID;

-- Create a new course
-- SELECT * FROM Course;
INSERT INTO Course VALUES
(1131, 'Al Usage', 'CS 4010', '11:00 - 1:00', 'Building 2',
'Computer Science');

-- Generate list of students in the Puzzle Club
SELECT Student.sID, Student.Name
FROM Student, Part_Of
WHERE Student.sID = Part_Of.sID
AND Part_Of.cName = 'Puzzle Club';
```

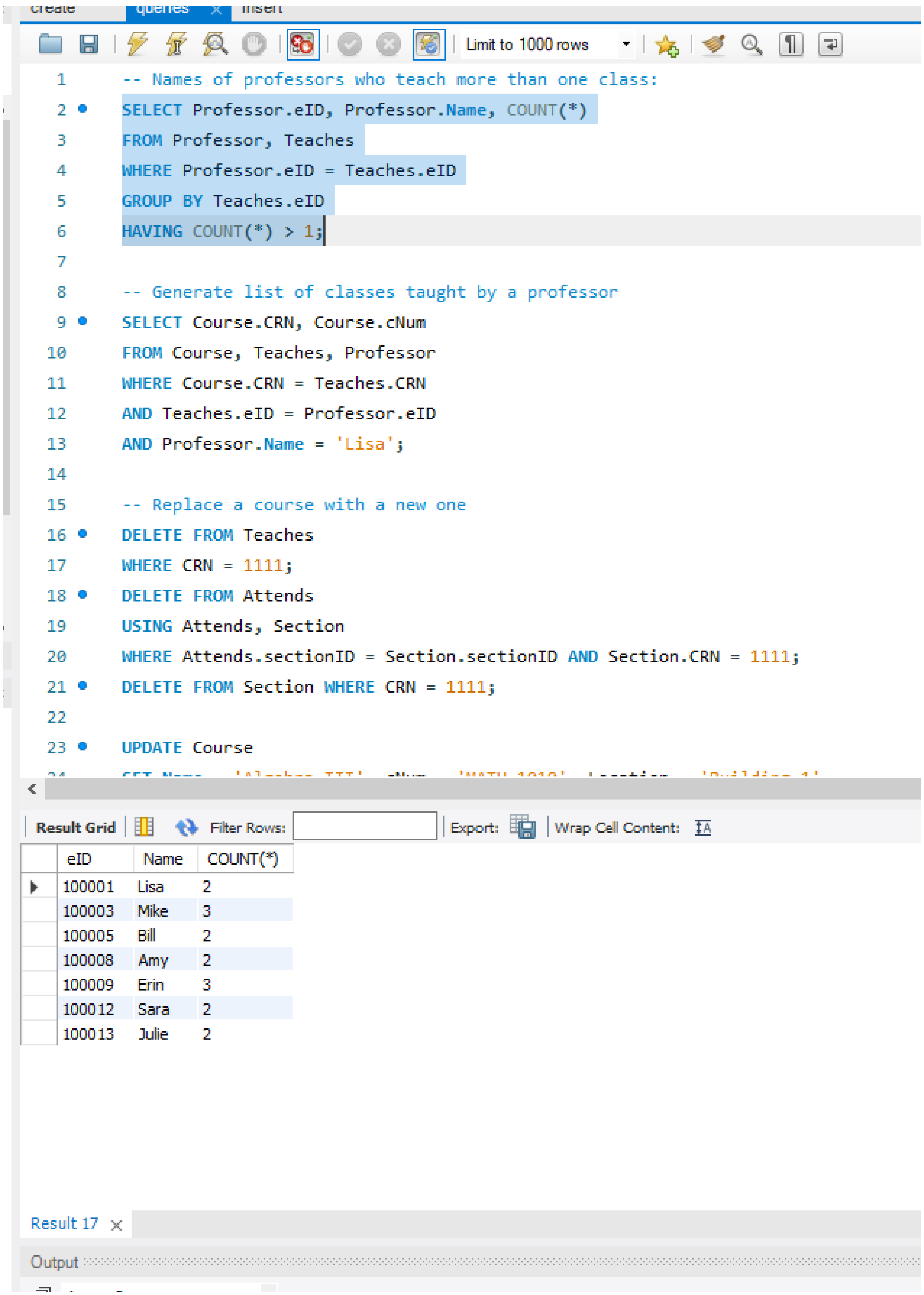
```
-- Query reserve record of Amenity 2
SELECT clubName, room, reserveTime
FROM Reserve
WHERE aName = 'Amenity 2';

-- Register a student to a section
SELECT * FROM Section, Attends WHERE Section.sectionID =
Attends.sectionID;
INSERT INTO Attends (sID, sectionID) VALUES
(2000004, 6);
```

Some Screenshots from Query Results



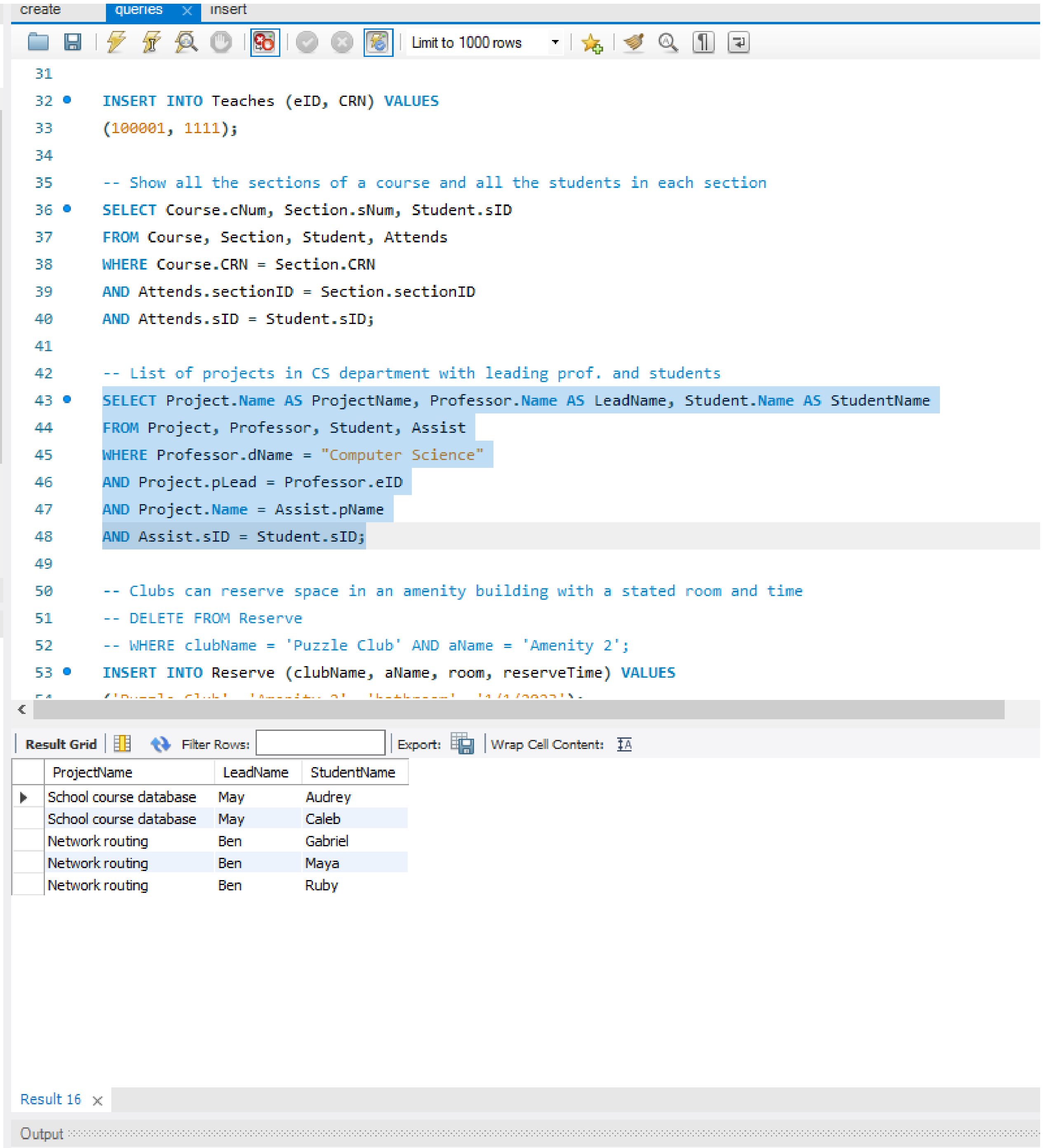
```sql

UPDATE Course
SET Name = 'Algebra III', cNum = 'MATH 1010', Location = 'Building 1',
Time = '9:00 - 11:00', Location = 'Building 2', dName = 'Mathematics'
WHERE CRN = 1111;
INSERT INTO Section (sNum, CRN, TA) VALUES
(1, 1111, 2000001),
(2, 1111, 2000002),
(3, 1111, 2000003);

INSERT INTO Teaches (eID, CRN) VALUES
(100001, 1111);

-- Show all the sections of a course and all the students in each section
SELECT Course.cNum, Section.sNum, Student.sID
FROM Course, Section, Student, Attends
WHERE Course.CRN = Section.CRN
AND Attends.sectionID = Section.sectionID
AND Attends.sID = Student.sID;

-- List of projects in CS department with leading prof. and students
SELECT Project.Name AS ProjectName, Professor.Name AS LeadName, Student.Name AS StudentName
FROM Project, Professor, Student, Assist
WHERE Professor.dName = "Computer Science"
```

cNum	sNum	sID
MATH1002	1	2000004
MATH1002	1	2000067
MATH1002	2	2000005
MATH1002	2	2000068
MATH1002	3	2000004
MATH1002	3	2000006
MATH1002	3	2000069
MATH 1003	1	2000007
MATH 1003	1	2000070
MATH 1003	2	2000008
MATH 1003	2	2000071
MATH 1003	3	2000009
MATH 1003	3	2000072
MATH 2001	1	2000010



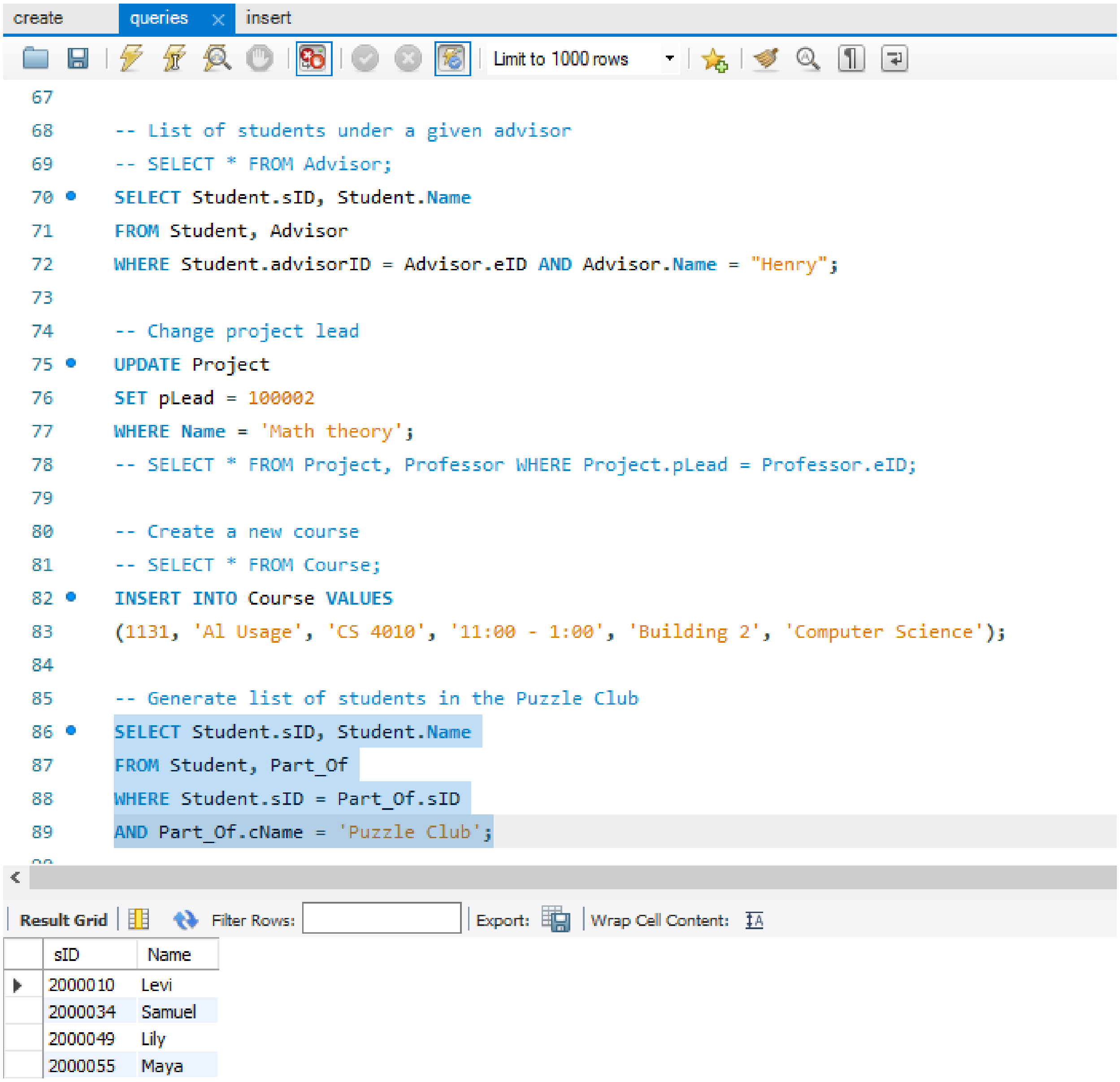
```sql
-- Names of professors who teach more than one class:
SELECT Professor.eID, Professor.Name, COUNT(*)
FROM Professor, Teaches
WHERE Professor.eID = Teaches.eID
GROUP BY Teaches.eID
HAVING COUNT(*) > 1;

-- Generate list of classes taught by a professor
SELECT Course.CRN, Course.cNum
FROM Course, Teaches, Professor
WHERE Course.CRN = Teaches.CRN
AND Teaches.eID = Professor.eID
AND Professor.Name = 'Lisa';

-- Replace a course with a new one
DELETE FROM Teaches
WHERE CRN = 1111;
DELETE FROM Attends
USING Attends, Section
WHERE Attends.sectionID = Section.sectionID AND Section.CRN = 1111;
DELETE FROM Section WHERE CRN = 1111;

UPDATE Course
SET Name = 'Algebra III', cNum = 'MATH 1010', Location = 'Building 1'
```

eID	Name	COUNT(*)
100001	Lisa	2
100003	Mike	3
100005	Bill	2
100008	Amy	2
100009	Erin	3
100012	Sara	2
100013	Julie	2



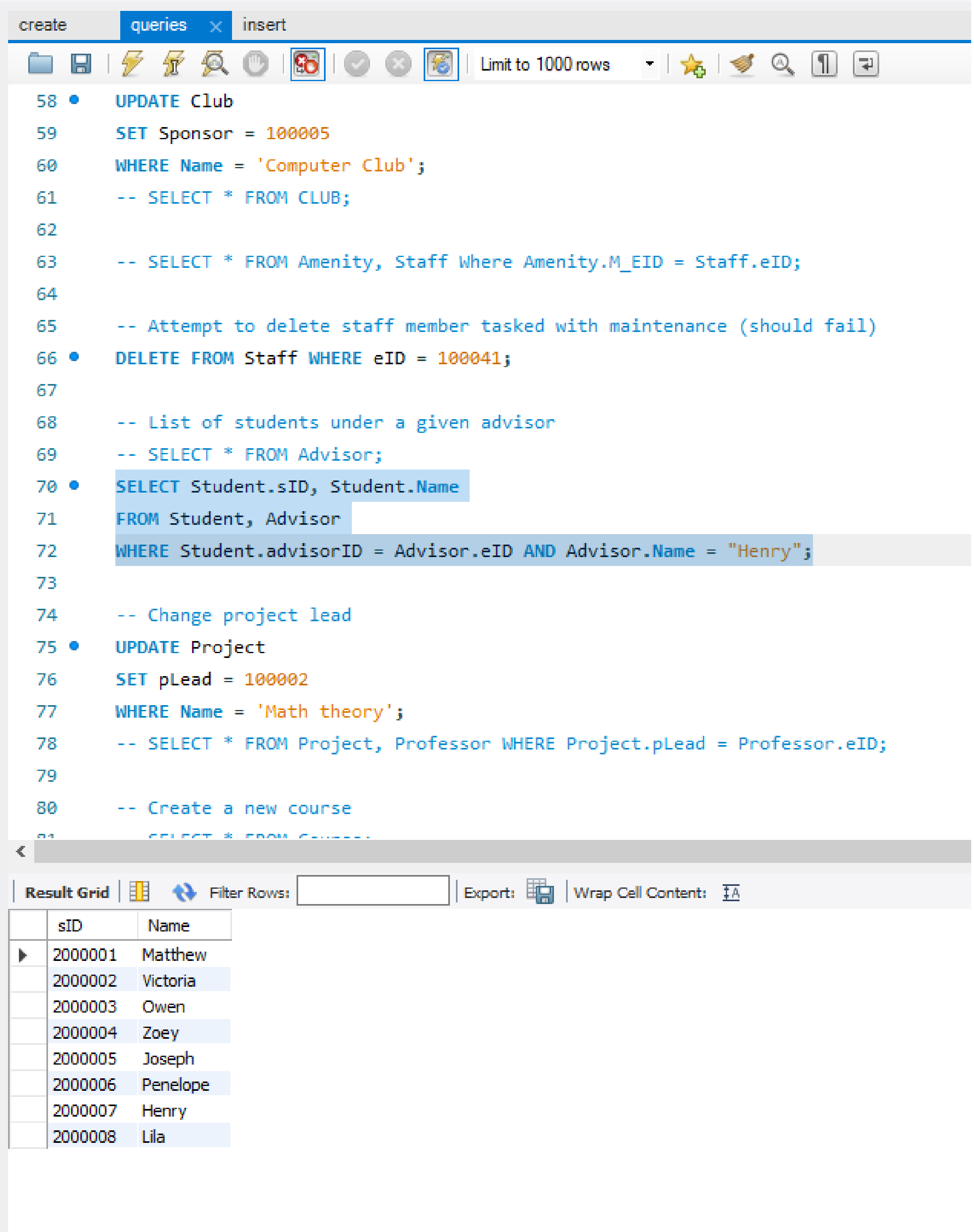
```sql

INSERT INTO Teaches (eID, CRN) VALUES
(100001, 1111);

-- Show all the sections of a course and all the students in each section
SELECT Course.cNum, Section.sNum, Student.sID
FROM Course, Section, Student, Attends
WHERE Course.CRN = Section.CRN
AND Attends.sectionID = Section.sectionID
AND Attends.sID = Student.sID;

-- List of projects in CS department with leading prof. and students
SELECT Project.Name AS ProjectName, Professor.Name AS LeadName, Student.Name AS StudentName
FROM Project, Professor, Student, Assist
WHERE Professor.dName = "Computer Science"
AND Project.pLead = Professor.eID
AND Project.Name = Assist.pName
AND Assist.sID = Student.sID;

-- Clubs can reserve space in an amenity building with a stated room and time
-- DELETE FROM Reserve
-- WHERE clubName = 'Puzzle Club' AND aName = 'Amenity 2';
INSERT INTO Reserve (clubName, aName, room, reserveTime) VALUES
('Puzzle Club', 'Amenity 2', 'bathroom', '1/1/2023');
```

ProjectName	LeadName	StudentName
School course database	May	Audrey
School course database	May	Caleb
Network routing	Ben	Gabriel
Network routing	Ben	Maya
Network routing	Ben	Ruby



```
-- List of students under a given advisor
-- SELECT * FROM Advisor;
SELECT Student.sID, Student.Name
FROM Student, Advisor
WHERE Student.advisorID = Advisor.eID AND Advisor.Name = "Henry";

-- Change project lead
UPDATE Project
SET pLead = 100002
WHERE Name = 'Math theory';
-- SELECT * FROM Project, Professor WHERE Project.pLead = Professor.eID;

-- Create a new course
-- SELECT * FROM Course;
INSERT INTO Course VALUES
(1131, 'Al Usage', 'CS 4010', '11:00 - 1:00', 'Building 2', 'Computer Science');

-- Generate list of students in the Puzzle Club
SELECT Student.sID, Student.Name
FROM Student, Part_Of
WHERE Student.sID = Part_Of.sID
AND Part_Of.cName = 'Puzzle Club';
```

sID	Name
2000010	Levi
2000034	Samuel
2000049	Lily
2000055	Maya



```sql
UPDATE Club
SET Sponsor = 100005
WHERE Name = 'Computer Club';
-- SELECT * FROM CLUB;

-- SELECT * FROM Amenity, Staff Where Amenity.M_EID = Staff.eID;

-- Attempt to delete staff member tasked with maintenance (should fail)
DELETE FROM Staff WHERE eID = 100041;

-- List of students under a given advisor
-- SELECT * FROM Advisor;
SELECT Student.sID, Student.Name
FROM Student, Advisor
WHERE Student.advisorID = Advisor.eID AND Advisor.Name = "Henry";

-- Change project lead
UPDATE Project
SET pLead = 100002
WHERE Name = 'Math theory';
-- SELECT * FROM Project, Professor WHERE Project.pLead = Professor.eID;

-- Create a new course
-- SELECT * FROM Course;
```

sID	Name
2000001	Matthew
2000002	Victoria
2000003	Owen
2000004	Zoey
2000005	Joseph
2000006	Penelope
2000007	Henry
2000008	Lila