Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

I. System analysis and ER model

Problem Statement: The registrar at Baruch College is committed to efficiently manage the course registration system to ensure a seamless academic experience for all students. The process begins with a designated registration starting date and concludes with a registration deadline, allowing students to enroll within the specified timeframe via the CUNYfirst website. Each course accommodates multiple students, and students have the flexibility to enroll in multiple courses as per their academic requirements. While each course is led by a single professor, instructors may teach multiple courses across the curriculum. For continued enrollment in courses, students must maintain a clear account status without any holds and fulfill all prerequisite requirements for their desired classes. This ensures academic progression and readiness for course material. Students are encouraged to seek guidance from advisors regarding their class enrollments and general course advice. Each advisor possesses a unique identification, complemented by their first and last names. Advisors may counsel multiple students, with each student eligible for up to one advising appointment per semester. Furthermore, the course registration system incorporates various elements to enhance its functionality, including course sections with unique schedules and capacities, careful classroom assignments based on factors like capacity and accessibility, course evaluations for ongoing improvements, and consideration of course attributes and restrictions to cater to diverse student needs and academic goals. Additionally, a registration status tracker feature is proposed, enabling real-time monitoring of registration progress, from course selection to enrollment confirmation. This enhancement aims to offer transparency and visibility into the registration process, empowering students to prioritize courses aligned with their scheduling needs. Ultimately, these efforts contribute to a robust registration system supporting student success and academic excellence at Baruch College.Through comprehensive management and attention to detail, the registrar ensures a robust course registration system that supports student success and academic excellence at Baruch College.

Assumptions:

- Students clear their account of holds before enrolling in courses
- There is a finite amount of seats per class
- There is only one registrar office

- Students must asses an evaluation for each professor at the end of the semester
- Students cannot enroll in classes before the starting date or after the deadline date, without exceptions
- Students can enroll in multiple courses per semester
- Professors can teach multiple sections simultaneously
- Students may only enroll in classes virtually
- Students must take a minimum of 9 credits per semester (part-time)
- Students must take a minimum of 12 credits per semester (full-time)
- Students may only take a maximum of 18 credits per semester
- Each student has a unique EMPLID
- A professor must teach at least one section
- Each course has a unique code
- Students have met the prerequisites of the class they are trying to enroll in
- All classes share the same start date and deadline date for enrollment for their respective semester
- Students must schedule an appointment with one advisor to help register for classes
- A course can have different sections.
- A classroom can host one section and course at a time
- A section can only use one classroom

Relationship Sentences:

- One student may enroll in multiple classes (within the set boundaries of credits)
- A class can host many students (within the limit of available seats)
- A professor can teach one or more classes per semester, simultaneously
- There is only one registrar office/website portal
- Each class can only be taught by one professor (at a time)
- Each advisor could advise multiple students and a student can have up to one appointment per semester
- A student has access to one evaluation per course
- Each section can be hosted in one classroom but a classroom can host multiple sections
- The registration system facilitates the enrollment of multiple students in multiple courses. However, each course registration process is managed by one registration system.

Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

Entities and Attributes

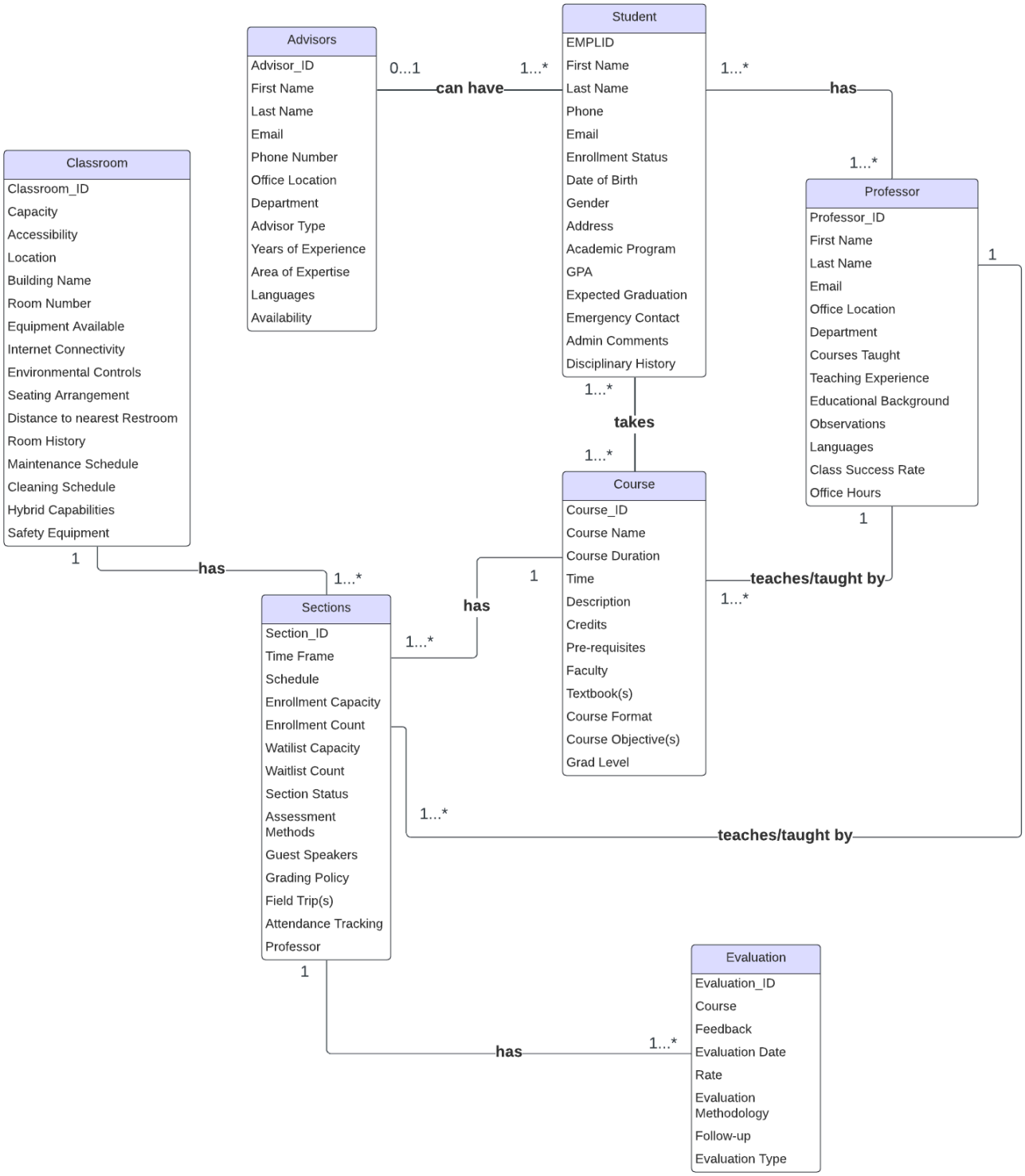
Our First Iteration

Entities	Attributes
Advisors	Advisor_ID, First name, Last name, Email, Phone number, Office location, Department, School, Advisor type (Academic Advisor, Career Advisor), Years of experience, Area of expertise, Languages spoken, Availability
Courses	Course_ID, Course name, Course duration, Description, Credits, Pre-requisites, Faculty, Textbooks required, Course format, Objectives, Grad Level (undergraduate, graduate)
Student	EMPLID, First name, Last name, Date of birth, Gender, Address, City, State, Zip Code, Email, Academic program, Enrollment status (Part-Time/Full-Time), Academic Program, GPA, Expected graduation date, Emergency contact information, Comments, Disciplinary history(if any)
Instructor	Instructor_ID, First name, Last name, Email, Office location, Department, Courses taught, Years of teaching experience, Education background, Teaching observations, Languages spoken, Class success rate, Office hours
Classroom	Classroom_ID, Building name, Room number, Capacity, Accessibility features, Location, Equipment available, Internet connectivity (wired, wireless), Environmental controls (temperature, lightning), Seating arrangement, Distance to restrooms, Room usage history, Maintenance schedule, Cleaning schedule, Hybrid capabilities, Safety equipment
Evaluation	Evaluation_ID, Feedback, Evaluation date, Rate, Evaluation methodology (online, paper-based), Follow-up, Evaluation type (Course evaluation, Instructor evaluation)
Sections	Section_ID, Time frame, Schedule,Enrollment capacity, Enrollment count, Waitlist capacity, Waitlist count, Section status(open, closed), Assessment methods,

Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

	Guest speakers, Grading policy, Field trips

Entity Relation Diagram



Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

II. Logical modeling and Normalization

Logical Modeling

Entities	Attributes
Advisors	Advisor_ID, First name, Last name, Email, Phone number, Office location, Department, School, Advisor type (Academic Advisor, Career Advisor), Years of experience, Area of expertise, Languages spoken, Availability.
Courses	Course_ID, Course name, Course duration, Description, Credits, Pre-requisites, Faculty, Textbooks required, Course format, Objectives, Grad level (undergraduate, graduate), Instructor_ID(fk).
Student_Course	Student_Course_ID, EMPLID(fk), Course_ID(fk)
Student	EMPLID, First name, Last name, Date of birth, Gender, Address, City, State, Zip Code, Phone, Email, Academic program, Enrollment status (Part-Time/Full-Time), GPA, Expected graduation date, Emergency contact information, Comments, Academic advising appointment history, Course registration history, Disciplinary actions (if any), Advisor_ID(fk)
Student_Instructor	Student_Instructor_ID, EMPLID(fk), Instructor_ID(fk)
Instructor	Instructor_ID, First name, Last name, Email, Office location, Department, Courses taught, Years of teaching experience, Education background, Teaching observations, Languages spoken, Student success rate, Office hours, Rate
Classroom	Classroom_ID, Building name, Room number, Capacity, Accessibility features, Location, Equipment available, Internet connectivity (wired, wireless), Environmental controls (temperature, lightning), Seating arrangement, Distance to restrooms, Room usage history, Maintenance schedule, Cleaning schedule, Hybrid capabilities, Safety equipment
Evaluation	Evaluation_ID, Feedback, Evaluation date, Rate, Evaluation methodology (online, paper-based), Follow-up, Evaluation type

	(Course evaluation, Instructor evaluation), Section_ID(fk)
Sections	SectionID, Time frame, Schedule,Enrollment capacity, Enrollment count, Waitlist capacity, Waitlist count, Section status(open, closed), Assessment methods, Guest speakers, Grading policy, Field trips, Attendance tracking system, Instructor_ID(fk), Classroom_ID(fk), Course_ID(fk)

Normalization process:

Assumptions about normalization exercise:

- All tables not explicitly shown to be normalized are considered normalized after our previous iteration
- Intended to demonstrate how we worked through this process

Step 1:

The objective is to normalize tables to minimize redundancy and inconsistencies, making it easier to update and maintain databases. This process helps improve our data's integrity, reduce storage space and enhance our query efficiency. Our aim is also to consolidate as much information as possible to reduce the total amount of tables.

Step 2:

Identifying the entities that have transitive or partial dependencies that need to be normalized:

Relation	Attributes
Advisors	Advisor_ID, First name, Last name, Email, Phone number, Office location, Department, School, Advisor type (Academic Advisor, Career Advisor), Years of experience, Area of expertise, Languages spoken, Availability.

Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

Student	EMPLID, First name, Last name, Date of birth, Gender, Address, City, State, Zip Code, Phone, Email, Academic program, Enrollment status (Part-Time/Full-Time), GPA, Expected graduation date, Emergency contact information, Comments, Academic advising appointment history, Course registration history, Disciplinary actions (if any), Advisor_ID(fk)

Step 2:

And, we normalized this...

1NF:

Relation	Attributes
Advisors	Advisor_ID, First name, Last name, Email, Phone number, Office location, Department, School, Advisor type (Academic Advisor, Career Advisor), Years of experience, Area of expertise, Languages spoken, Availability.

2NF:
Partial Dependency:

- **Department** → School

Advisors(Advisor_ID, First Name, Last Name, Email, Phone Number, Office Location, Advisor Type, Years of Experience, Area of Expertise, Languages Spoken, Availability

Advisor_Department(Department, School)

* with the removal of the partial dependency, there remain no partial or transitive dependencies therefore making this iteration also in **3NF**

Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

1NF:

Student	EMPLID, First name, Last name, Date of birth, Gender, Address, City, State, Zip Code, Phone, Email, Academic program, Enrollment status (Part-Time/Full-Time), GPA, Expected graduation date, Emergency contact information, Comments, Academic advising appointment history, Course registration history, Disciplinary actions (if any), Advisor_ID(fk)

2NF:
Partial Dependency:

- **Zip Code** → City, State

  * this dependency and deduction operates on the assumption that two different locations can have the same address as it pertains to “street/apartment/house address”

Student(EMPLID, First Name, Last Name, Date of Birth, Gender, Phone, Email, Address, Academic Program, Enrollment Status, GPA, Expected Graduation Date, Emergency Contact Information, Comments, Academic Advising History, Course Registration History, Disciplinary Actions, Advisor_ID)

Student_Zip(ZipCode, City, State)

* with the removal of the partial dependency, there remain no partial or transitive dependencies therefore making this iteration also in **3NF**

Ali Kasimoglu
Gioconda Prada

III. Physical database

SQL codes that helped us create our database with their primary keys and foreign keys:

```
CREATE TABLE Advisors (
   AdvisorID INT NOT NULL,
   FirstName VARCHAR(45) NULL,
   LastName VARCHAR(45) NULL,
   Email VARCHAR(45) NULL,
   PhoneNumber VARCHAR(20) NULL,
   OfficeLocation VARCHAR(100) NULL,
   Department VARCHAR(100) NULL,
   AdvisorType VARCHAR(50) NULL,
   YearsOfExperience INT NULL,
   AreaOfExpertise VARCHAR(200) NULL,
   LanguagesSpoken VARCHAR(200) NULL,
   Availability VARCHAR(100) NULL,
   PRIMARY KEY (AdvisorID)
);

CREATE TABLE Courses (
   Course_ID INT NOT NULL,
   Course_name VARCHAR(100) NULL,
   Course_duration VARCHAR(50) NULL,
   Description TEXT NULL,
   Credits INT NULL,
   Pre_requisites TEXT NULL,
   Faculty VARCHAR(100) NULL,
   Textbooks_required TEXT NULL,
   Course_format VARCHAR(50) NULL,
   Objectives TEXT NULL,
   Grad_level VARCHAR(20) NULL,
   Instructor_ID INT,
   PRIMARY KEY (Course_ID),
   FOREIGN KEY (Instructor_ID) REFERENCES Instructor(Instructor_ID)
);
```

Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

```
CREATE TABLE Instructor (
    Instructor_ID INT NOT NULL,
    First_name VARCHAR(50) NULL,
    Last_name VARCHAR(50) NULL,
    Email VARCHAR(100) NULL,
    Office_location VARCHAR(100) NULL,
    Department VARCHAR(100) NULL,
    Courses_taught TEXT NULL,
    Years_of_teaching_experience INT NULL,
    Education_background TEXT NULL,
    Teaching_observations TEXT NULL,
    Languages_spoken VARCHAR(200) NULL,
    Student_success_rate INT NULL,
    Office_hours VARCHAR(100) NULL,
    Rate INT NULL,
    PRIMARY KEY (Instructor_ID)
);

CREATE TABLE Students (
    EMPLID INT NOT NULL,
    First_name VARCHAR(50) NULL,
    Last_name VARCHAR(50) NULL,
    Date_of_birth DATE NULL,
    Gender VARCHAR(10) NULL,
    Address VARCHAR(100) NULL,
    Phone VARCHAR(20) NULL,
    Email VARCHAR(100) NULL,
    Academic_program VARCHAR(100) NULL,
    Enrollment_status VARCHAR(20) NULL,
    GPA INT NULL,
    Expected_graduation_date DATE NULL,
    Emergency_contact_information VARCHAR(200) NULL,
    Comments TEXT NULL,
    Academic_advising_appointment_history TEXT NULL,
    Course_registration_history TEXT NULL,
    Disciplinary_actions TEXT NULL,
    AdvisorID INT,
```

```
    PRIMARY KEY (EMPLID),
    FOREIGN KEY (AdvisorID) REFERENCES Advisors(AdvisorID)
);

CREATE TABLE Student_Instructor (
    Student_Instructor_ID INT NOT NULL,
    EMPLID INT,
    Instructor_ID INT,
    PRIMARY KEY (Student_Instructor_ID),
    FOREIGN KEY (EMPLID) REFERENCES Students(EMPLID),
    FOREIGN KEY (Instructor_ID) REFERENCES Instructor(Instructor_ID)
);

CREATE TABLE Classroom (
    Classroom_ID INT NOT NULL,
    Building_name VARCHAR(100) NULL,
    Room_number VARCHAR(20) NULL,
    Capacity INT NULL,
    Accessibility_features TEXT NULL,
    Location VARCHAR(100) NULL,
    Equipment_available TEXT NULL,
    Internet_connectivity VARCHAR(50) NULL,
    Environmental_controls TEXT NULL,
    Seating_arrangement TEXT NULL,
    Distance_to_restrooms VARCHAR(50) NULL,
    Room_usage_history TEXT NULL,
    Maintenance_schedule TEXT NULL,
    Cleaning_schedule TEXT NULL,
    Hybrid_capabilities TEXT NULL,
    Safety_equipment TEXT NULL,
    PRIMARY KEY (Classroom_ID)
);

CREATE TABLE Evaluation (
    Evaluation_ID INT NOT NULL,
    Feedback TEXT NULL,
    Evaluation_date DATE NULL,
```

```
    Rate INT NULL,
    Evaluation_methodology VARCHAR(50) NULL,
    Follow_up TEXT NULL,
    Evaluation_type VARCHAR(50) NULL,
    SectionID INT,
    PRIMARY KEY (Evaluation_ID),
    FOREIGN KEY (SectionID) REFERENCES Sections(SectionID)
);

CREATE TABLE Sections (
    SectionID INT NOT NULL,
    Time_frame VARCHAR(100) NULL,
    Schedule TEXT NULL,
    Enrollment_capacity INT NULL,
    Enrollment_count INT NULL,
    Waitlist_capacity INT NULL,
    Waitlist_count INT NULL,
    Section_status VARCHAR(10) NULL,
    Assessment_methods TEXT NULL,
    Guest_speakers TEXT NULL,
    Grading_policy TEXT NULL,
    Field_trips TEXT NULL,
    Attendance_tracking_system VARCHAR(100) NULL,
    Instructor_ID INT,
    Classroom_ID INT,
    Course_ID INT,
    PRIMARY KEY (SectionID),
    FOREIGN KEY (Instructor_ID) REFERENCES Instructor(Instructor_ID),
    FOREIGN KEY (Classroom_ID) REFERENCES Classroom(Classroom_ID),
    FOREIGN KEY (Course_ID) REFERENCES Courses(Course_ID)
);
```

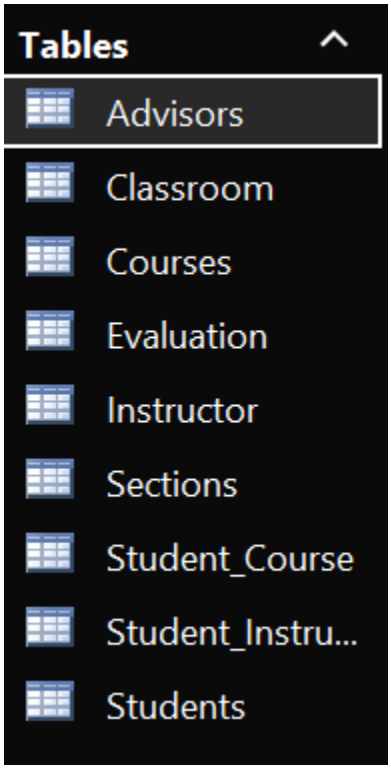
Decided to add a few more attributes

```
ALTER TABLE Advisors
        ADD School VARCHAR(100);
```

Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

```
ALTER TABLE Students
        ADD COLUMN Zip_code VARCHAR(10) NULL,
                City VARCHAR(100) NULL,
                State VARCHAR(50) NULL;
```

After setting up the tables, here's how the database schema looks now:



Ali Kasimoglu
Gioconda Prada
CIS 9340 - Group Project

With the Relationship View feature in Database Tools, we can visualize the connections (foreign keys) between the tables. Let's take a look at it now.