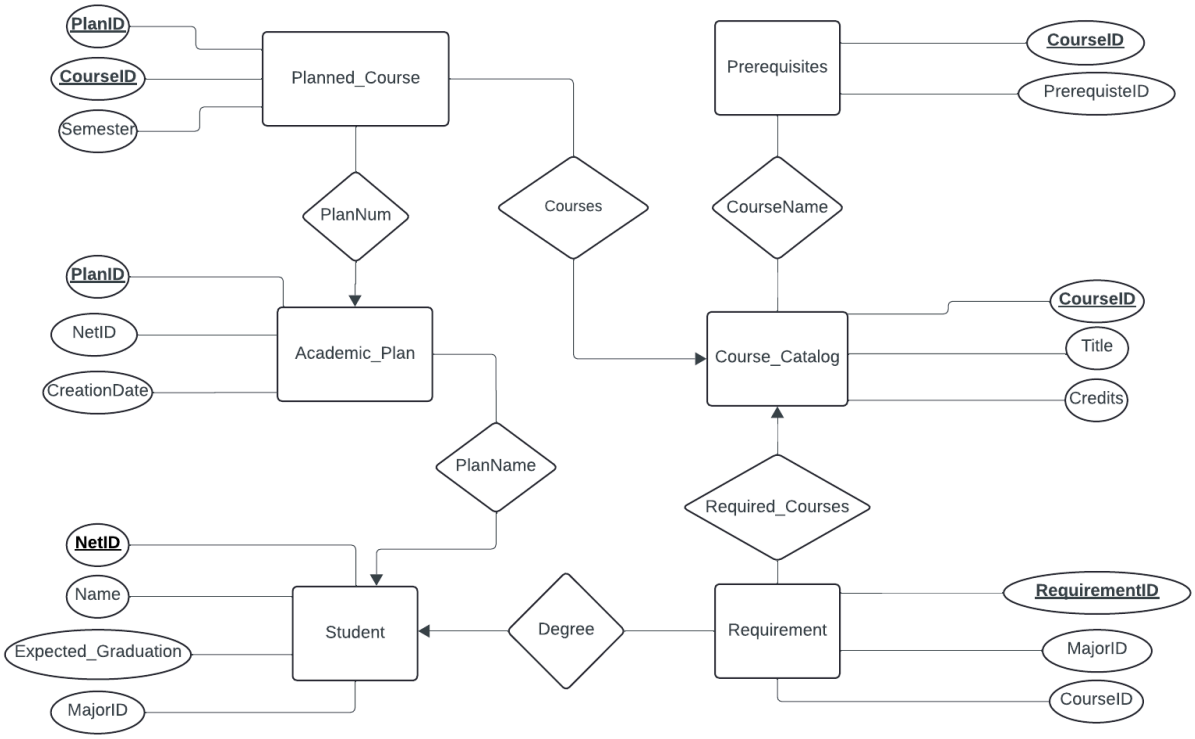
Stage 2



Entities and Attributes

1. Student:
Each student has a unique NetId(PK), defined by attributes such as `Name`, `Expected_Graduation`, and `MajorID`. Since a student has multiple attributes, and is also associated with other entities (e.g., academic plan), it is modeled as an entity rather than as an attribute of an academic plan.

2. Academic_Plan:
Represents a set of planned courses and uniquely defined as a PlanID(PK) for a student with Attributes: `CreationDate`, `PlanName` . An academic plan involves multiple attributes that describe the plan itself and is linked to multiple courses, which makes it more suitable as a distinct entity.

3. Planned_Course:
This entity refers to a course that has been planned in an academic plan. The attributes are `CourseID`, `Semester`, 'PlanID'(PK). The reason why this is a separate entity is because a course can be planned multiple times for different semesters and different students, it needs to have its own set of attributes and relationships.

4. Course_Catalog:
This entity holds information about courses offered by the institution. With Attributes: `CourseID`(PK), `Title`, `Credits`. The catalog contains multiple courses that are shared across multiple students and plans, hence it is more appropriate as a separate entity.

5. Prerequisites:
This entity represents prerequisite relationships among courses. Attributes: `PrerequisiteID`, `CourseID`(PK). Prerequisites involve many-to-many relationships between courses, which makes it necessary to have this as a distinct entity rather than an attribute.

6. Requirements:
This entity represents requirements relationships among majors and courses. Attributes: `RequirementID`(PK), `CourseID`, `MajorID`. Requirements involve one-to-many relationships between courses, which makes it necessary to have this as a distinct entity rather than an attribute and one-many relationships with students since each major can have multiple requirements.

Relationships and Cardinalities

1. Student and Academic_Plan (One-to-Many):
A student can have multiple academic plans, but each plan belongs to one student.This allows for students to have different plans, perhaps for different degrees or revisions.

2. Academic_Plan and Planned_Course (One-to-Many):
An academic plan can include multiple planned courses, but each planned course belongs to one specific academic plan. It reflects the structure of planning courses within a single degree plan.

3. Course_Catalog and Prerequisites (Many-to-Many):
A course can have multiple prerequisites, and a prerequisite can be applicable to multiple courses. A separate relationship is required to accurately model this many-to-many association.

4. Student and Requirements (One-to-Many):
A student has one major but a major can have multiple students. This is consistent with how majors are typically assigned, where multiple students are part of the same major.

5. Requirements and Courses(One-to-Many):
A course can be required by multiple majors, but a major will only require unique courses.

Entities List:

- Student (NetID, Name, ExpectedGrad, MajorID)
- Course (CourseID, Title, Credits)
- Requirement (RequirementID, MajorID, CourseID)
- AcademicPlan (PlanID, StudentID, CreationDate)
- PlannedCourse (PlanID, CourseID, Semester)
- CoursePrerequisite (CourseID, PrerequisiteID)

FUNCTIONAL DEPENDENCIES

1) Student
   NetID → Name, ExpectedGrad, MajorID
2) Course_Catalog
   CourseID → Title, Credits
3) Requirement
   RequirementID → MajorID,CourseID
4) Academic Plan
   PlanID → NetID, CreationDate
5) Planned_Course
   PlanID, CourseID → Semester
6) Prerequisite
   CourseID → PrerequisiteID

NORMALIZED USING 3NF

1. Student (NetID, Name, ExpectedGrad, MajorID)

   PK: NetID FK: MajorID

2. Course_Catalog (CourseID, Title, Credits)

   PK: CourseID

3. Requirement (RequirementID,MajorID, CourseID)

   PK: RequirementID FK: MajorID FK: CourseID

4. Academic_Plan (PlanID, NetID, CreationDate)

   PK: PlanID FK: NetID

5. Planned_Course (PlanID, CourseID, Semester)

   PK: (PlanID, CourseID) FK: PlanID FK: CourseID

6. Prerequisite (CourseID, PrerequisiteID)

   PK: (CourseID) FK: CourseID FK: PrerequisiteID

The following normalized dependencies adheres to 3NF for the following reasons:

- Every non prime attribute in the entity is fully dependent on the primary key
- This ensures that there are no partial dependencies
- There are also no transitive dependencies since all of the attribute is dependent on the primary key, this ensures that there are no dependencies from non-prime to another non-prime attribute

The dependencies above are 1NF because all of the attribute values are atomic (no lists or tuple values). The dependencies are also 2NF because the entity attributes are all fully dependent on the primary key. And finally the dependencies are 3NF because there are no transitive dependencies (non-prime to another non-prime attribute dependencies)

Relational Schema

Student(
  NetID: INT [PK],
  Name: VARCHAR(100),
  Expected_Graduation: DATE,
  MajorID: VARCHAR(10)
)

Course_Catalog(
  CourseID: VARCHAR(10) [PK],
  CourseName: VARCHAR(100),
  Credits: INT
)

Requirement(
  RequirementID: VARCHAR(10) [PK],
  MajorID: VARCHAR(10) [FK to Student.MajorID],
  CourseID: VARCHAR(10) [FK to Course.CourseID]
)

AcademicPlan(
  PlanID: INT [PK],
  StudentID: INT [FK to Student.StudentID],
  CreationDate: DATE
)

PlannedCourse(
  PlanID: INT [FK to AcademicPlan.PlanID, PK],
  CourseID: VARCHAR(10) [FK to Course.CourseID, PK],
  Semester: INT
)

CoursePrerequisite(
  CourseID: VARCHAR(10) [FK to Course.CourseID, PK],
  PrerequisiteID: VARCHAR(10) [FK to Course.CourseID]
)