Relational Schema:

User_Table(net_id:VARCHAR(15) [PK], username:VARCHAR(15), user_password:VARCHAR(30), first_name:VARCHAR(30), last_name:VARCHAR(30), graduation_month:INT, graduation_year:INT)

Course_Information(course_code:VARCHAR(15) [PK], course_name:VARCHAR(100), credit_hours:INT)

Courses_Completed_By_User(net_id:VARCHAR(15) [PK, FK to User_Table.net_id], course_code:VARCHAR(15) [PK, FK to Course_Information.course_code], semester_taken:VARCHAR(15)[PK], year_taken:INT[PK])

Courses_Planned_By_User(net_id:VARCHAR(15) [PK, FK to User_Table.net_id], course_code:VARCHAR(15) [PK, FK to Course_Information.course_code], semester_planned:VARCHAR(15)[PK], year_planned:INT[PK])

Course_GPA_By_Instructor(instructor:VARCHAR(100) [PK], course_code:VARCHAR(15) [PK, FK to Course_Information.course_code], percentage_a:DECIMAL(6,3), percentage_b:DECIMAL(6,3), percentage_c:DECIMAL(6,3), percentage_d:DECIMAL(6,3), percentage_f:DECIMAL(6,3))

GenEds_Dataset(course_code:VARCHAR(15) [PK, FK to Course_Information.course_code], gened_requirement_fulfillment:VARCHAR(100))

Prerequisite(course_code:VARCHAR(15) [PK, FK to Course_Information.course_code], prerequisite_course_code:VARCHAR(15) [PK, FK to Course_Information.course_code], requirement_group_id:INT [PK])

Concurrent_Enrollment(course_code:VARCHAR(15) [PK, FK to Course_Information.course_code], concurrent_enrollment_course_code:VARCHAR(15) [PK, FK to Course_Information.course_code])

Entity Design Justification & 3NF

Entity	Justification for Attributes	Why It's an Entity, Not an Attribute	3NF Compliance
User Table	The attributes (net_id, username, first_name, user_password, Graduation Month, Graduation Year) all directly relate to to describe a single, unique user. Each piece of information is a property of a person using the system.	A user is a primary actor in the system. It has relationships with many other entities such as planned courses and completed courses. If "user" were just an attribute like a user name then we would have to repeat all the user's details for every course they plan or have taken, which makes it redundant.	1NF: Each attribute holds only one value. 2NF: The primary key is NetId; since it's a single unique key, all attributes fully depend on it as each NetId is assigned one singular student. 3NF: Each attribute depends only on the net_id. No attribute determines another attribute.
Course Information	All attributes (course_code, course_name, credit_hours) are essential details that are used to identify a single academic course.	A course is a central concept that represents a varied number of courses rather than being a single property. It can be planned by many users, have many prerequisites, and be taught by many instructors. This can't be an attribute because the course exists independently of any user or instructor. If it was simply an attribute,	1NF: Each of the attributes course_code, course_name, course_hour are one value. 2NF: The primary key is CourseCode so all attributes depend entirely on it 3NF: There are no transitive dependencies the attributes

		we would have to repeat this information in the many tables that reference it.	CreditHours and CourseName only depend on CourseCode
Courses Completed By User	The attributes NetId, Course Code, Semester Taken, Year Taken connect a specific User to the Course. These serve as a record for each of the users.	This is an associative entity for the many-to-many relationship between users and courses, as a user can complete many courses and a course can be taken by many users. This resolves it to a valid mapping as a junction table between those two tables.	1NF: All of the attributes NetId, Course Code, Semester Taken, Year Taken are one value. 2NF: This table satisfies it because the composite key (NetID, CourseCode) uniquely identifies each record, and both SemesterTaken and YearTaken attributes depend on both the full key rather than a part of it because we need to know what user took what course when. 3NF: Non-key attributes (SemesterTaken, YearTaken) depend only on the composite key, not on each other.

Courses Planned By User	The attributes NetId, Course Code, Year Planned and Semester Planned are details that describe a connection between the two entities User and the Course. This provides additional details of that relationship.	This is an associative entity for the many-to-many relationship between users and courses, as a user can plan many courses and a course can be planned by many users. This resolves it to a valid mapping as a junction table between those two tables.	1NF: The attributes NetId, Course Code, Year Planned and Semester Planned are all one value. 2NF: This table satisfies it because the composite key (NetID, CourseCode) uniquely identifies each record, and both SemesterPlanned and YearPlanned attributes depend on both the full key rather than a part of it because we need to know what user took plans to take what course when. 3NF: SemesterPlanned and YearPlanned depend only on (NetID, CourseCode).
Course GPA By Instructor	The attributes combine an Instructor and a Course Code to show the grade distribution of each instructor teaching a specific course aggregated by all	This is an associative entity for the many-to-many relationship between instructors and courses, as an instructor teaches many courses, and a course can be taught by many instructors. Thus, we need to make it an entity to store the grade	1NF: The instructor, course code, and each grade distribution are all one value. 2NF: The composite key is (Instructor, CourseCode). The grade distribution is specific to

	the semesters they taught it.	distribution that is specific to the combination of an instructor and a course to resolve it to a valid mapping.	the instructor and what course was taught. 3NF: The grade distribution values do not depend on each other.
GenEds Dataset	These attributes link a course code to the general education requirements it fulfills.	If we tried to make gened_requirement an attribute of Course_Information, we would create redundant rows for the same course, destroying 1 to 1 relationships with other tables.	1NF: Each record lists one course and one requirement that it fulfills. 2NF: The key CourseCode would determine the type of requirement that is fulfilled. 3NF: There are no other dependencies as the GenEdRequirementFulfillment depends only on the key CourseCode.
Prerequisite	The two attributes of this table are course_code and prerequisite_course_code. This tells us which course must be taken before	This associative entity is necessary for us to represent the many to many self referencing relationship. Otherwise, the course information table would have multiple entries per course for each of its prerequisites, which	1NF: The attributes Prerequisite Code and Course Code are one value. 2NF: The composite key (CourseCode, PrerequisiteCourseCode) uniquely identifies each relationship: what is the

	taking a particular course.	would make lookups to that table more complex, destroying the 1 to 1 relationship it has with other tables.	prerequisite for each course. 3NF: There are no non-key attributes so no transitive dependency can occur.
Concurrent Enrollment	The two attributes of this table are course_code and concurrent_enrollment_course_code. This tells us that two courses can be taken together to be eligible for both. These attributes point to the course code entity.	This associative entity is necessary for us to represent the many to many self referencing relationship. Otherwise, the course information table would have multiple entries per course that can serve as its concurrent enrollment, which would make lookups to that table more complex, destroying the 1 to 1 relationship it has with other tables.	1NF: Each record stores only one pairing of courses. 2NF: The composite key (CourseCode, ConcurrentEnrollmentCourseCode) uniquely defines each row: what is the possible concurrent enrollment for each course. 3NF: There are no non-key attributes so no transitive dependency can occur.

- **For each many to many Relationship, we added a junction table**

Relationship and its Cardinality

Relationship	**Cardinality**	**Cardinality Description**
User Table - Courses	Zero to Many	Each user (NetID) may have

Completed by User		completed zero or more courses because they could be an incoming freshman or current student.
Courses Completed By User - User Table	One To One	Every course completed by the user maps to one user record, since netID is unique to the user.
User Table - Courses Planned By User	Zero To Many	Each user can have zero or many planned courses because a user might have planned courses or chosen not to.
Courses Planned By User - User Table	One To One	Each planned course for a user is associated with just that user's record.
Course Information - Courses Completed By User	Zero to Many	Each course could have been completed by 0 users if it is new or many if it has been offered before.
Courses Completed By User - Course Information	One to One	Each completed course record refers to exactly one course taken.
Course Information - Courses Planned by the User	Zero To Many	Each course could have been planned by 0 users if no user planned to take the course or many if many planned to take it.
Courses Planned by the User - Course Information	One To One	Each planned course record refers to the exact course information of that planned course.
Course Information - GPA By Instructor	Zero To Many	If the course has not been offered before, it will not have an associated grade distribution. If the course has been offered before, it can have many GPAs depending

		on if different instructors offered it.
GPA By Instructor - Course Information	One To One	Each course code in the GPA instructor table refers to only one course in the course information table.
Course Information - GenEDs dataset	Zero to Many	Each course in the course information table can fulfill zero or many GenED requirements.
GenEds Dataset - Course Information	One to One	Every genED course maps to the exact course it is in the course information table.
Course Information - Concurrent Enrollment	Zero to Many	Each course is the concurrent enrollment course for zero or many courses. Each course can have zero or many concurrent enrollment courses. We assume that no course would have concurrent enrollment with two other courses at the same time but rather the option with either course.
Concurrent Enrollment - Course Information	One to One	Each concurrent enrollment course code maps back to that exact course information entry. Each course code can only be linked to one information row about that course code.
Course Information - Prerequisite	Zero to Many	Each course can have zero or many prerequisites.
Prerequisite - Course Information	One to One	Each course can be the prerequisite for zero or many courses. Each course can have zero or many prerequisites. Each course code can only be linked to one information row about

		that course code.