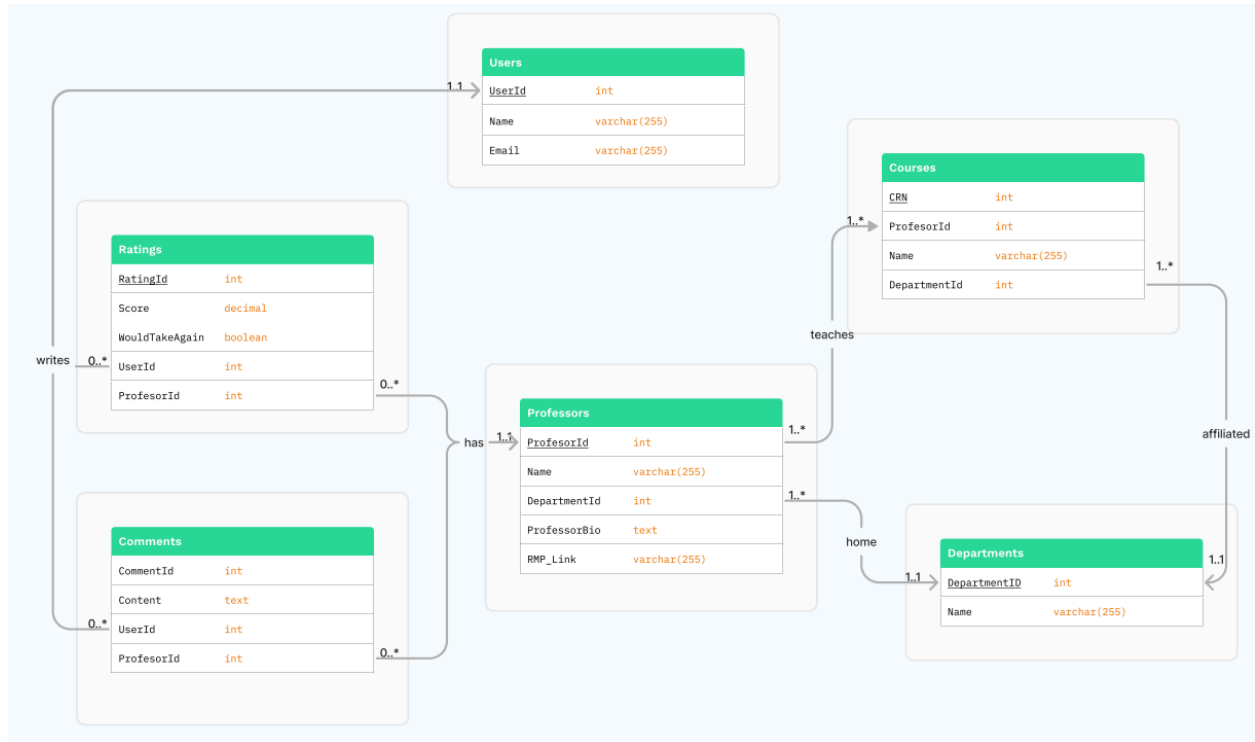


UML Diagram



Assumptions

Courses Entity:

Each course is uniquely identified by a Course Registration Number (CRN) and is associated with one professor and one department. Courses is an entity because it represents a distinct concept with attributes that describe it such as CRN. We need to see what course people are giving reviews on. A professor can teach many courses and courses can have multiple professors(many to many), and a department can offer many courses and each course has one department(many to one).

Departments Entity:

Each department is unique and can be associated with multiple professors and courses. This helps us filter by department when looking for courses or professors. Departments have a one to many relationship with courses and professors since both can only have one department.

Users Entity:

Users are individuals who can rate and comment on professors. This is important so that we can let a user comment and give a rating. Each user has a unique identity in the system. Users have a one to many relationship with comments and ratings since each comment and rating can only have one user that gives it.

Ratings Entity:

Users can provide multiple ratings, but each rating is linked to one professor and one user. Ratings is its own entity because each rating is important when generating a score to give the professor. The relationship is many to one from Ratings to both Users and Professors. A user can give multiple ratings, and a professor can receive multiple ratings, but ratings can only be for one professor.

Comments Entity:

Similar to ratings, users can post multiple comments, but each comment is about one professor and made by one user. It has a distinct relationship with Users and Professors so it needs to be its own entity. The relationship is many to one from Comments to both Users and Professors since users and professors can have multiple comments and comments can only be for one professor.

Professors Entity:

Professors are unique individuals associated with one department. They can teach multiple courses and have multiple ratings and comments. It has distinct relationships with Courses, Departments, Ratings, and Comments so it is its own entity. The relationship is many to one from Professors to Departments as a professor belongs to one department but departments can have many professors. With courses a professor can also teach many courses and a course can have multiple professors so that is many to many too.

Functional Dependencies

CRN->ProfessorId, Name, DeptID

DepartmentId->Name

ProfessorId->Name, DepartmentId, ProfessorBio,RMP_Link

UserID->Name, Email

RatingID->Score, WouldTakeAgain, ProfessorID, UserID

CommentID->Content, ProfessorID, UserID

We chose 3NF instead of BCNF decomposition due to its simplicity and a less complex database structure. Each table has a primary key that uniquely identifies its rows, and all other attributes in each table are fully functionally dependent on the primary key. The left side are all singletons and there are no redundancies or any need for further decomposition. The minimal bases are the same as the functional dependencies. So, every table is normalized in 3NF form.

Relationship Schema

Courses(CRN: INT [PK], ProfessorID: INT[FK], Name: VARCHAR, DepartmentID: INT [FK to Department.DepartmentID])

Department(DepartmentID: INT [PK],Name: VARCHAR)

Users(UserID: INT [PK], Name: VARCHAR, Email: VARCHAR)

Ratings(RatingID: INT [PK],Score: DECIMAL, WouldTakeAgain: BOOLEAN, ProfessorID: INT [FK to Professor.ProfessorID], UserID: INT [FK to User.UserID])

Comments(CommentID: INT [PK], Content: TEXT, ProfessorID: INT [FK to Professor.ProfessorID], UserID: INT [FK to User.UserID])

Professors(ProfessorID: INT [PK], Name: VARCHAR, DepartmentID: INT [FK to Department.DepartmentID], ProfessorBio: TEXT, RMP_Link: VARCHAR)