**Class**: CECS 323 Sec 01

**Project Name**: MongoDB Enrollment Project Design

**Team #**: 5

**Members**: Andrew York, Emily Hawkins, Devin Korthari, Yunjong Noh

| **Parent Table Name** | **Child Table Name** | **Multiplicity** | **Direction** | **Design Pattern** | **Comments** | **Rationale** |
| --- | --- | --- | --- | --- | --- | --- |
| departments | majors | 1 ..1 to 1 ..\* | Parent to Child | One-to-Many | Each department document will have an array of the major IDs it offers. | Departments have several majors that it offers and this will allow both department and majors to be queried separately |
| departments | majors | 1 ..1 to 1..\* | Child to Parent | Two-Way Referencing | Each course document will have the ID of its parent document. | This is so we can get what department the major is a part of when we have the major |
| departments | courses | 1 ..1 to 0 ..\* | Parent to Child | One-to-Many | Each department document will have an array of course IDs. | A department has many courses that it offers. |
| departments | courses | 1 ..1 to 0 ..\* | Child to Parent | Two-Way Referencing | Each course document will have the department abbreviation. | Courses are often identified with the combination of the course abbreviation and the course number so it would be nice to have them both appear if the courses document is queried and department names/abbreviations don’t appear to change often. |
| majors | studentmajors | 1 ..1 to 0 ..\* | Parent to Child | One-to-Many | Each major will have a list of declared-students’ major ID | One major can have at least a couple hundred students but less than thousand (Like CSULB) |
| majors | studentmajors | 1 ..1 to 0 ..\* | Child to Parent | Two-Way Referencing | Each studentmajor will have the major name, which is a unique identifier, derived from its parent document | Each student has a specific declaration date followed by a majors(or university) request. |
| students | studentmajors | 1 ..1 to 0 ..\* | Parent to Child | Denormalization from Many to One | The students document will have an array that contains the studentmajor ID and the major name per index | Each student will have mostly likely up to a few major (like major and minor) and it would be convenient to see the student’s major when they’re queried |
| students | studentmajors | 1 ..1 to 0 ..\* | Child to Parent | Two-Way Referencing | The studentmajors document will contain the ID of the student that has declared the major | This is in order to know which student declared which major. |
| courses | sections | 1 ..1 to 0..\* | Parent to Child | One-to-Many | The course document will have an array of section IDs that it keeps track of | One course can have multiple sections but a section cannot be associated with multiple courses |
| courses | sections | 1 ..1 to 0.. \* | Child to Parent | Denormalization from One-to-Many | The section document contains the department abbreviation and the course number from the course document it belongs to. | A section belongs to a course identified with a course number and its department abbreviation. |
| students | enrollments | 1 ..1 to 0 ..\* | Parent to Child | Denormalization from Many to One | Students will maintain an array of objects containing the enrollment ID, department abbreviation, and course number | A student can be enrolled in multiple courses. It would be nice to be able to see the courses that a student is enrolled in when querying the student. |
| students | enrollments | 1 ..1 to 0 ..\* | Child to Parent | Two-Way-Referencing | The enrollment document will contain the student ID | This is so we can identify who is enrolling in a particular section |
| sections | enrollments | 1 ..1 to 0 ..\* | Parent to Child | One-to-Many | Sections will have an array of enrollment IDs | A section can have multiple students enrolled |
| sections | enrollments | 1 ..1 to 0 ..\* | Child to Parent | Denormalization from One-to-Many | Enrollments will contain the department abbreviation, course number, section number, semester, and year of the section document | All the attributes uniquely identify the section and it seems unlikely that the data for a specific section will change once the semester starts. |
| enrollments | passfail | 1 ..1 to 0 ..1 | These two categories of enrollments are complete and disjoint so we can create subschemas and have data be validated against the subschemas, using “oneOf” to ensure that the data for enrollment is valid according to one of the subschemas. | | | |
| enrollments | lettergrade | 1 ..1 to 0 ..1 |