

Enrollment Project Team 5

Parent	Child	Multiplicity	Direction	Design Pattern	Comments	Rational
departments	courses	1..1 to 1..*	Parent to child	One to many	One department encompasses many courses	departments will have a course list field with names that reference course instances. This makes it easy to reference the departments courses but extra work needs to be done when deleting a course from the database (delete the course from the course field)
courses	departments	1..* to 1..1	Child to parent	Many to one	One course can only be affiliated with one department	There is a department field in the course collection that stores the name of the department affiliated with the given course instance. Again, this makes it easy to reference which department this course belongs to
courses	sections	1..1 to 1..*	Parent to child	One to many	One course can have many sections	There will be a section list field with strings that reference section instances. This makes it convenient to reference but extra work needs to be done when deleting a section (delete the section from this list field)

sections	courses	1..* to 1..1	Child to parent	Many to One	A section can only belong to a single course	One section can only have one course affiliation, so sections will have a course field that stores the name of the given course instance
sections	enrollments	1..1 to 1..*	Parent to child	One to squillions	The enrollment instances themselves are subclasses on enrollments, (incomplete, disjoint) and either PassFail or LetterGrade	The enrollments collection is the parent collection and PassFail and LetterGrade are children of that class that share all fields except for their single unique field
enrollments	sections	1..* to 1..1	Child to parent	Denormalization from one to squillions	Enrollments has semester, section_year from sections in order to enforce a uniqueness constraint	This denormalization is accepted as the fields that are redundant are constantly used to make sure that there aren't any violations of the uniqueness constraints given
students	enrollments	1..1 to 1..*	Parent to child	One to squillions	One student can have many different enrollments across different sections	A student can potentially have a large amount of enrollments over the course of their college career, so one to squillions is preferred over one to many
enrollments	students	1..1 to 1..*	Child to Parent	Squillions to one	Many enrollments can belong to a single student	The enrollment collection itself refers to its associated student document
students	student_majors	1..1 to 1..1	Parent to Child	One to one	A student can only have one declared major This relationship is done to connect students to majors collection	The student collection itself doesn't have any field that references student_major but it passes to student_major
student_majors	students	1..1 to 1..1	Child to Parent	One to one	A student_major can only reference one student	student_majors has a field that holds the ObjectID of the student

majors	student_majors	1..1 to 1..1	Parent to Child	One to Many	A major can have many different student_majors This relationship is done to connect majors and students collection	Many students share the same major, this is what this relationship is trying to show
student_majors	majors	1..1 to 1..1	Child to Parent	Many to one	Many student_majors can have the same major	student_major has a field that references a specific major instance
departments	majors	1..1 to *..1	Parent to child	One to many	Every department has many majors affiliated with itself	The majors are kept in a list, makes it easy to reference but extra work needs to be done when deleting a major to also delete it from this list
majors	departments	1..* to 1..	Child to Paren	Many to One	There are many majors that are encompassed within under a single department	Majors are referenced in department as a list of majors that the department is affiliated with