Overview

In this tutorial we will

- Create two additional entities to the ERD that we created in *Systems Analysis and Design, Learning Module Series #5, Creating an Elementary Level ERD*
- Customize the fields/columns of these entities, and assign primary keys
- Break the many-to-many relationship between Student and Course _Section_ into two separate one-to-many relationships using an Intersection entity, namely, the Registration entity
- Adjust the diagram symbols (such as the crows foot symbol)
- Link the entities together, and
- Decipher the semantics of the created ERD

The first new entity (the Student entity) will contain a list of students who are taking some of the sections associated with courses in our original diagram. Our goal is to be able to store (and accurately query) the attribute “final_grade”, which is a functional dependency of Student and Section together:

{ (student, section) \(\rightarrow\) final_grade }

Since a given student may be taking many sections; while any given section may be associated with many students, there is no unambiguous way of identifying which set of records in one table belongs to a single record in the other.

Each record in the Student Entity must be unique, that is, it must represent a unique (or individual) student. To store the attribute final_grade in the **Student Entity** would mean that every row in the entity would contain a student_name along with an associated final_grade. But if a given student is registered for more than one
section, he will not receive a single, total grade; rather he will receive many grades. The Student Entity is therefore inadequate in terms of being able to store all of his grades.

Similarly, each record in the Course_Section Entity must represent a unique (or individual) course section title. To store the attribute final_grade in the Course_Section Entity would mean that every row in the table would contain a course_section_name along with an associated final_grade. But how would we identify which student in a class of twenty is associated with this grade, for no one course section has merely one grade assigned for that section; rather it has multiple associated grades corresponding to each student that completed the class. The Course_Section Entity then, is also inadequate in terms of being able to store student grades.

It comes out that neither entity can effectively contain the attribute final_grade.

As a result, the relational model does not support the direct implementation of many-to-many relationships, such a this one, or any other; and the only solution is to break the many-to-many relationship into a pair of binary, one-to-many relationships. This is achieved by introducing a third table into the pattern. These “third” tables have various names, but are most commonly referred to as either “intersection”, “composite”, or “cross-reference” tables.

This new Intersection entity (the Registration entity) is utilized to eliminate the many-to-many relationship that exists between students and course sections. The ID’s of the Student and Course_Section entities, respectively, are both placed into the intersection entity and combined into a single key -- a composite Primary Key. The student_ID is then placed into the Course_Section entity as a Foreign Key, and likewise, the course_section_ID is placed into the Student entity also as a Foreign Key. Additionally, the Registration entity will contain the attribute final_grade.

In plain English, the Registration entity is needed because it mediates between students and course sections. Each semester, a student may create multiple registrations for access to multiple course sections, and each semester, any given course section will likely have many different students registered for it.

Creating the Student & Registration Entities

Create the Student & Registration Entities. Customize Columns, Assign primary keys and set the stage for our newly created Intersection Entity.
1. Under Database Properties, under Categories, click on Definition to name the entity **Student**.

2. Click on Columns and under the Physical Name column, enter the following attributes: **student_ID** (primary key), **course_section_ID**, **last_name**, **first_name**, **address**, **city**, **state**, **zip** and **ssn**. Finally uncheck the PK check box for **course_section_ID** to remove its “Primary Key” designation.

3. Repeat steps 1 and 2 for the Registration entity. Its columns should be **student_ID** (primary key), **course_section_ID** (primary key) and **final_grade**.

4. Click on the Course_Section entity, and uncheck the PK check box for **student_ID** to remove its “Primary Key” designation.
At this point we should have three entities: **Course Section**, **Student** and **Registration**.

*The Course entity is not displayed here since even though it exists in the original ERD I, it will not be necessary to use it in this tutorial.*
1. From the Menu Bar, click on Database and select Options… Document from the cascading drop down menu to confirm that under the Relationship tab, Relationships and Crows Feet are checked. Click OK.

![Database Document Options](image)

**Linking the Entities Together**

6. From the Shapes pane, drag the Relationship shape to hover over the Course_Section entity and drop it when the Visio red box appears around it. Now connect the unattached end of the Relationship shape to the Registration entity – again dropping it only when the red box appears around it.

7. Under Database Properties… Categories … Miscellaneous select the Identifying radio button under “Relationship Type”.

8. ![Diagram](image)
9. From the Shapes pane, drag the Relationship shape and drop onto any available space on the workspace grid. Grab the “crows foot” end of the connector, and drag it to hover over the Registration Entity. When the Visio red box appears, drop it onto the Registration Entity.

6. Now grab the other end of the connector, and drag it to hover over the Student Entity. When the Visio red box appears, drop it onto the Student Entity.

Notice that under Database Properties, “Relationship type” is automatically set to Identifying.
You should now have an ERD that looks like this:

Deciphering the symbols

The Course_Section entity contains a unique ID for all course-sections and the Student entity contains a unique ID for all students. Each record in the Registration entity represents a registration of a student and his chosen course-section. Each combination is uniquely identified by a composite ID that references a single record (or section) in the Course_Section entity, as well as a single record (or student) in the Student entity.

Between Course-Sections and Registrations

- Reading from left to center, a course section can have zero or many registrations, but
- Reading center to left, a registration if, it exists, must belong to one and only course section.

Between Students and Registrations

- Reading from right to center, a student can have zero or many registrations, but
- Reading center to right, a registration, if it exists, must belong to one and only one student.
This Completes

Systems Analysis and Design
Learning Module Series #6

Document version

Advanced Issues in Entity Relationship Diagrams