

Entity Relationship Diagrams

ERD

Revisiting PEDAC

When creating a database-backed system we consider this part of our Data Structure

When modeling our systems, we need to represent real-world things in a format that lends itself to being stored in a relational database. One of our jobs as software developers is understanding the business process and the business problem well enough to accurately model the current problem and create a structure that can bend and change to future requirements.

ERD is a way to represent the structure of our business entities in a way that allows us to model our real-world data in database tables and relationships.

When working on a business problem, we should be looking for statements that reveal the types and structure of data we want to manage.

- Keep track of students, the courses they are taking, and the course's teacher.
- Record the customer's order, the items on each order, and which of the customer's addresses to ship to
- Record the author, time of creation, and contents of the blog post. Also, record comments, including who commented and when

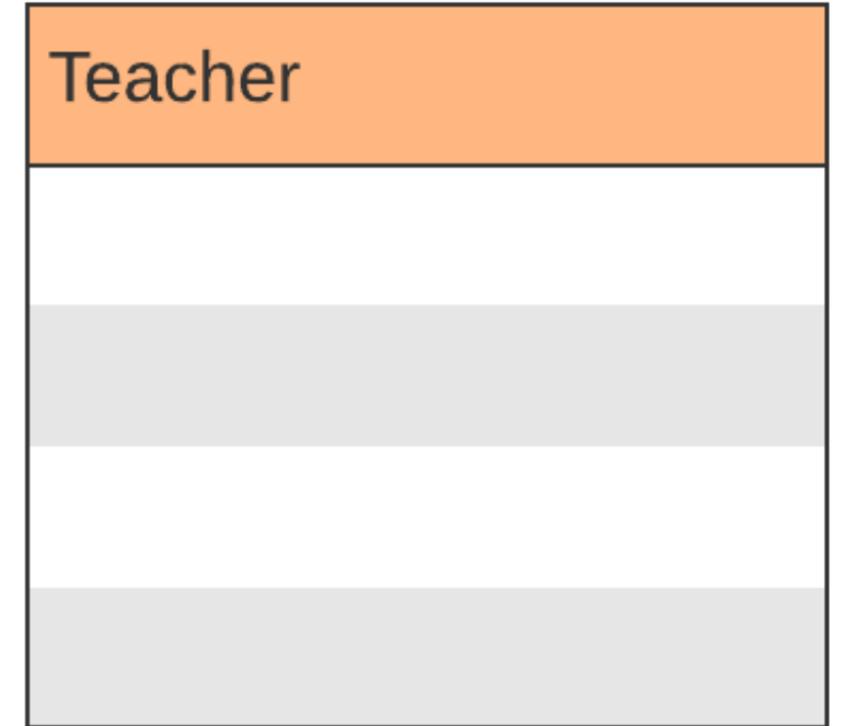
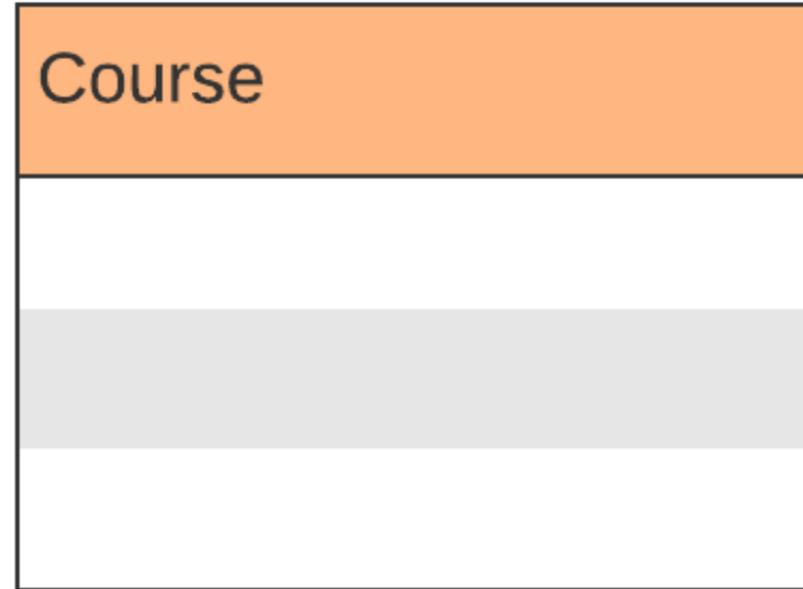
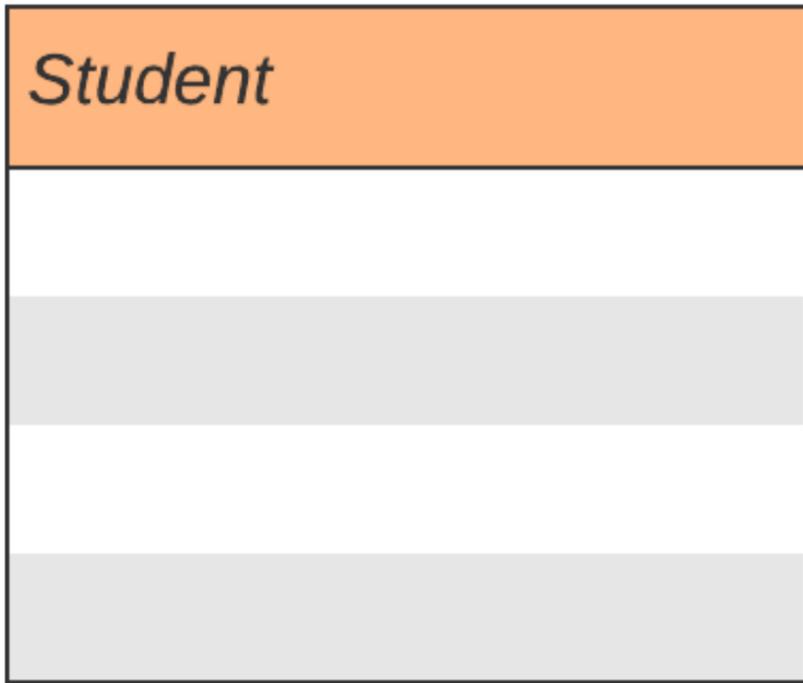
Within these descriptions, we are looking for different types of words to help guide us. We should be looking for:

- Nouns: a thing, such as a student, a customer, or an order.
- Verbs: an action, such as enroll or ship.
- Adjectives: a describing word such as posted or approved.

Components of an ERD

The first component of an ERD is an entity. These typically represent the significant *nouns* of our system.

"Keep track of students, the courses they are taking, and the course's teacher."



Attributes

Next, we will determine the attributes of each of these nouns.

These would be things like

- "the student's name"
- "the student's age"
- "the student's birthday"
- "the course's name"
- "the course's description"

<i>Student</i>
<i>Name</i>
<i>Age</i>
<i>Birthday</i>

Course
Name
Description

Teacher
Name

Naming Things

You'll notice that all the attributes are **singular** named and store **a single value**. If we had a multiple-value attribute of an entity, say multiple phone numbers for a Teacher, we'd create another entity to track those.

Data Types

In addition to the names of the attributes, we may also capture the type of data the attribute represents.

- Name is text
- Age is a number/integer
- Birthday is a date

These data types will become column types eventually in our database.

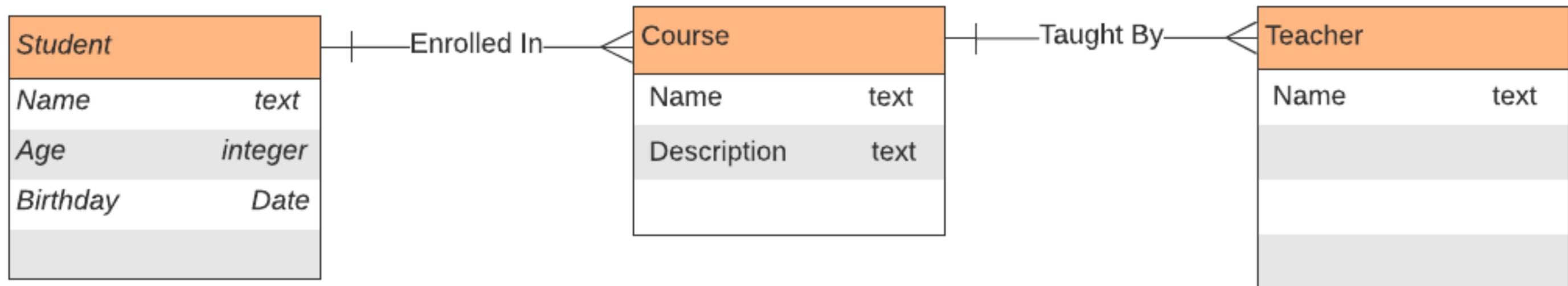
<i>Student</i>	
<i>Name</i>	<i>text</i>
<i>Age</i>	<i>integer</i>
<i>Birthday</i>	<i>Date</i>

Course	
Name	text
Description	text

Teacher	
Name	text

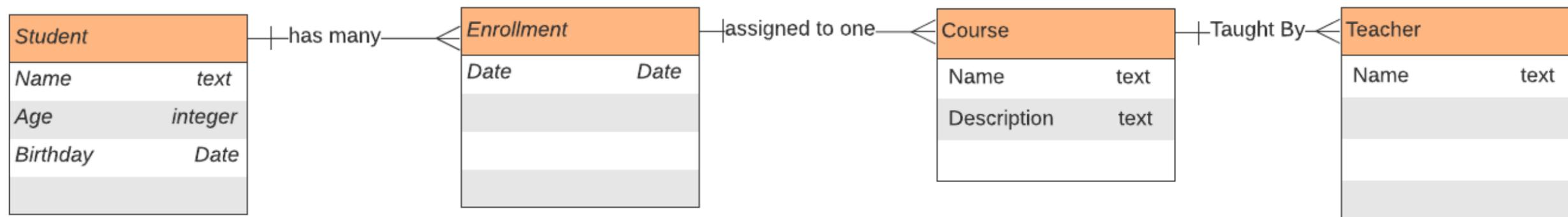
Relationships

After identifying the entities and their attributes, we can identify the relationships



Some relationships take more work (and entities)

If we determine that a relationship, in this case, *enrolled in* needs extra data stored along with it, say the date the student enrolled, we'll create another entity to hold it.



Translating to our database

- If we take the time to map out the entities, attributes, and relations in an ERD, we can begin to see the data structure in our application.
- Entity Relationship Diagrams also serve as good documentation for your project and allow you to review with the various *stakeholders* since the diagrams are also accessible for non-developers to understand (perhaps with a bit of help).