

706  
80

Jeremy Neal  
CMSC 461 - Databases  
11/10/2014  
Sleeman

### Project - Phase 3

The attached diagram shows the conversion of the entity-relationship model from Phase 2 to the relational model.

Several considerations were made in the design of this system. The strong entities from the ER diagram were taken and created into the main tables in the relational model. This included the User class, books, courses, universities, etc. In general, these items corresponded to objects of some kind, whether it was a person or a data point. Once these entities were accounted for, other entity sets were taken into consideration, and the relationships between them were implemented. While transforming our models from ER to relational, composite attributes were reduced to individual values as necessary. For example, names were often tracked as a composite attribute with a subattribute of first name and last name. These were translated to individual values.

In the ER diagram, there is specialization in the User model. To show this in the relational model, I chose to create a User base class that corresponded to all the users in the system, students and employees alike. The IDs of these entries were then used to link that user to their role, whether it was a student or employee. This seemed to provide a sufficient amount of specialization, and the use of more specific subclasses seemed unnecessary. The main values associated with these models don't change given further specificity; that is, we don't any more information about an admin or superadmin than we do about a customer service representative. Further specialization would only increase redundancy.

This is generally how relationships were described in the relational model. Relationships were shown using the primary key of a model (usually an ID value). For example, student users always have a cart. This cart can become an order. For the uses of our system, it worked well enough to retrieve these models using the ID of the student they are assigned to. A student doesn't need to have an order, but this relationship doesn't enforce such a condition.

One major design consideration was how to uniquely identify books. There were three attributes that, in conjunction, provide a satisfactory means of identifying books uniquely. These were the title, ISBN and ISBN-13 attributes. However, querying a database using three attributes can be a bit troublesome. It was decided to remove those values from the Book model, create a table called Book ID, and link those three values to a unique ID for each book. This way, those values are still accessible, but there is also a convenient means of uniquely identifying book selections.