Grade: 91/100

Jeremy Neal
CMSC 461 - Databases
10/27/2014
Sleeman

Project - Phase 2

The attached diagram lays out a high level design for the Book Fetch database application. Within, the relationships between various entities are defined.

A large part of the design was rigidly determined by the requirements listed in the document. However, there are some portions of this design which may warrant explanation.

One of the more obvious features is the generalization and specialization of users in the database. This was a rather straightforward way to reduce redundancy. While users have different abilities and responsibilities, the key attributes of most users overlap. This served as the best way to describe the users.

Additionally, it is worth noting that the distinction between a customer service user and an administrator is a single value, which could be either a numeric value or a string. The reasoning here is that while customer service users and administrators have very different responsibilities, the data stored about them is not very different. This serves as a simple and succinct way to describe the different users. Also, user privileges are outside the scope of this document, so the distinction is clear enough.

The relationship between employees and tickets was generalized to “processes”. This is mostly for the sake of simplicity. Employees have different responsibilities in regards to tickets, but the distinction was considered to be outside the scope of this design document. This ultimately is an issue of privileges.

There were some values that I chose to represent as values in the entity, while others used a relationship. This decision was based on the cardinality of the value itself. For example, categories of books were linked as a relation, because there is no way of knowing what all of the possible values could be. This is the same for subcategories of books. Conversely, the shipping type of orders was kept as an attribute of the entity. The reason for this is that we have a fixed number of possible options (standard, 1 day, 2 day). Creating another entity to hold these three static values would be a waste of space and possibly a performance issue. Other attributes in other entities also follow this convention, such as language, format and purchase types of books.

In the order entity, credit card name could have been split into a multivalued attribute, but for this particular entity, there seemed to be no reason to do so. In credit card transactions, the whole name is always used.