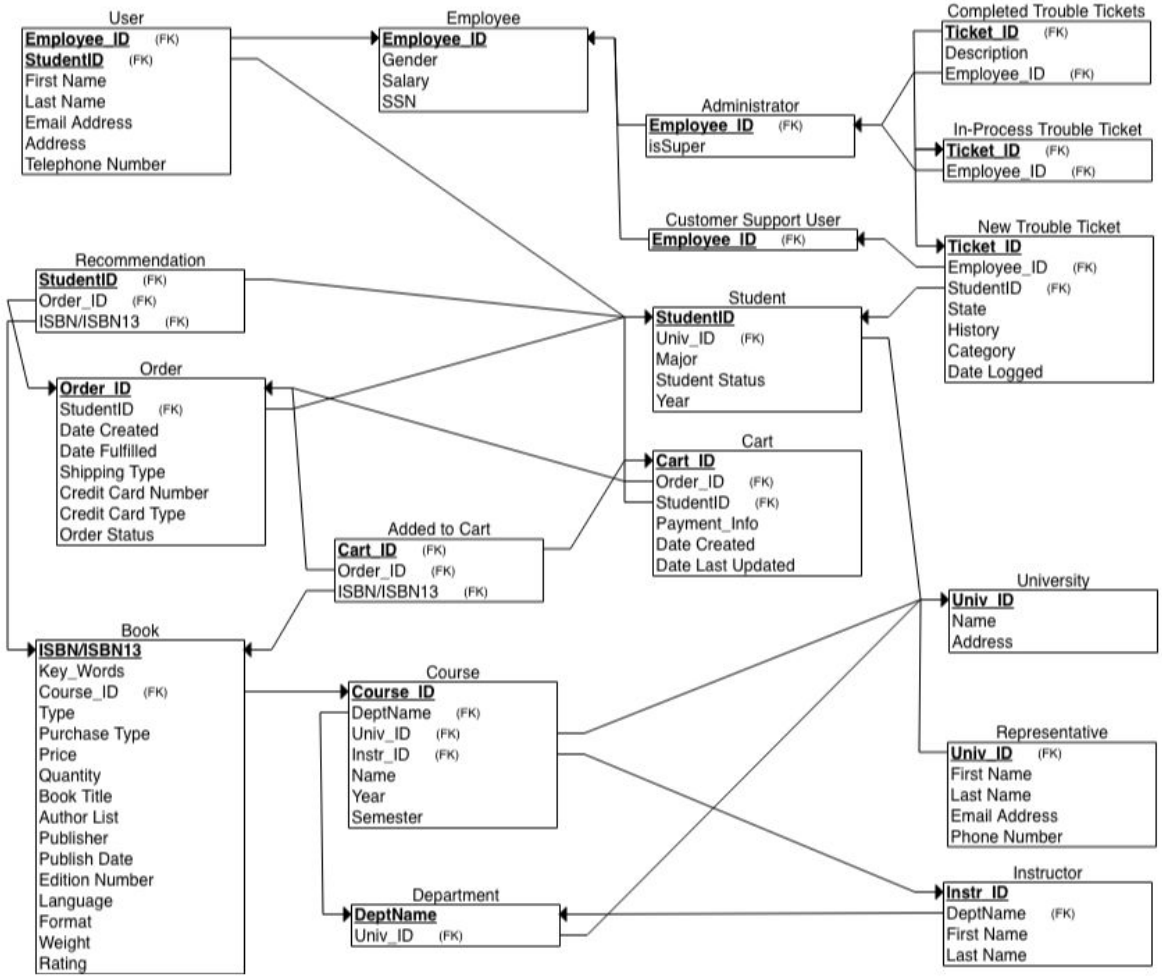
James Williams
CMSC 461
Project Phase 3
4/2/18

Relational Schema:



Write Up:

In the relation schema, I switched up the design in small ways that I initially hadn't thought of before while still keeping things relatively straight forward. Compared to my ER Diagram in phase 2, everything from there is here (minus the relationships) and a few new relations have been added to make things more accessible by query.

The first major change Was the generalization of the users in the database. I chose to generalize the Students, Customer Support Users, and the Administrators by creating a User table that stores all their generic information such as first name, last name, email, address, etc. In addition Customer Support Users and Administors have also been generalized into an Employees table. The reasoning behind using generalization is because if a query had to be done on all of the users, or all of the employees, no jons would have to been done in order to do so. This makes the data more accessible for querying.

The other major change was specialization, or adding more subclasses to make things simpler. For example I gave University representatives their own relation, I added two more relations to the trouble ticket system, and I made an 'Added to Cart' relation.

The reason I made the representative's relation was because it felt less relevant to the university data. Considering how different the data is, I considered it worthy to be placed in a different relation for specialization purposes. This way queries made for just the university info can be made without also getting the representatives info. Similarly queries made for just the representative's info can be made without getting the universities information.

The reason I made the trouble tickets in three relations was for the ease on security. Since all trouble tickets now aren't all in the same relation, making sure that an admin doesn't touch "new" tickets or a customer support user doesn't touch "in-process" tickets is now simpler. Since the different types of tickets are in different relations, it will be easier to restrict access to different types of tickets. However the harder part will be ensuring the transfer of tickets between relations. However I think the trade off of work is potentially worth it. No guarantees this will stay permanently though if I find anything that makes this approach significantly more difficult.

Lastly the reason I added the 'Added the Cart' relation was to get rid of some of the density in the cart relation. If a user were to add many books into their cart, a list of books attribute in the cart relation might become a little too overpopulated. Rather than that, any single book that is added to any cart will be put into this relation, along with the id of the cart that it was added to. Once again, this may not be permanent if this proves to not be that bad of a problem.