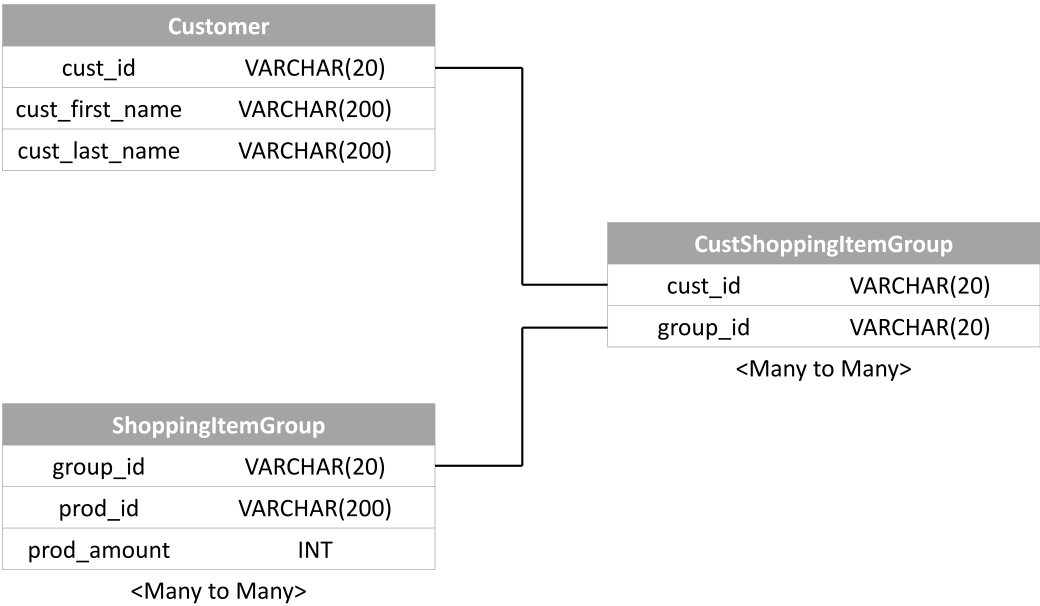


Additional Questions

Question 1

We want to give customers the ability to create lists of products for one-click ordering of bulk items. How would you design the tables, what are the pros and cons of your approach?

I will create a group of database tables as the following schema:



A table **Customer** will store the customer information. Another table **ShoppingItemGroup** will be a **many-to-many** relationship table for storing the selected products in a certain group. The third **CustShoppingItemGroup** will be storing the **many-to-many** relationship between Customer and his/her created list of products. The pro for this approach would be the simplicity of the implementation. The relationship is clear and the SQL command is easy to be constructed to obtain the data for a certain group of product for a user. The con might be the performance overhead while querying if all the table become bigger.

Question 2

If Shipt knew exact inventory of stores, and when facing a high traffic and limited supply of particular item, how do you distribute the inventory among customers checking out?

For those customer buying the same item, we design a queuing system for them. The fist customer arrived could obtain a system created token. The system will process the request for the fist arriving user with a valid token by first check the validity of the token. Then, the system will assign a lock for the item and only the current accepted request has the right to change the amount of that item. The other request will be pending within the queuing system. After the modification is completed, the finishing request will release the lock and the system will assign it to the next request in the queue.