Project Reflection Report

Our database project began with the creation of a schema with entities like Customer, Product, Orders, etc. and relations like Places (Customer to Orders), Includes (Orders to Products), etc. We started with the vision to create a marketplace interface to allow users to browse products and buy products they like. We kept the schema such as to avoid redundancy as much as possible. The functionalities that were thought of initially, were: addition and deletion of user accounts (buyer and merchant), creation of orders, inclusion of products in an order, etc. However, as can be seen from our initial schema, we did not have the total price of an order. As we progressed through the project, we realized that to create a trigger like waiving off the delivery fee if the order amount exceeds a threshold, we needed to have the total amount of the order in the table. Hence, this column `totalAmount` was later added and the schema was accordingly updated. Also, to enable the possibility of deleting records from the entities with the primary key, we applied constraints on the foreign keys referring to these primary keys. Apart from this, to meet the reasonable requirement that every time we create an order, we should not manually assign the ID to that order, we included the auto-increment feature on the primary key of Orders entity (ID).

The interface that we created fulfilled all the basic requirements of any e-marketing website and can be worked upon to transform into a state-of-the-art website through small changes. A major aspect that we could not achieve in our project was to create a predicted degree of satisfaction for a particular product for a user, which we intended to attain by using text analysis of the reviews they had previously given to the products matching on criteria like brand, type of product, price, etc. The feature sounds quite possible yet inventive, aiming to personalizing the “reviews” feature of most e-marketing websites. But the catch with such a feature is that we need to have enough data on the user transactions. Given the fact that we had to fill in most of the tables with dummy data, including Reviews, which could hardly provide sensible results upon applying NLP, we skipped the part. However, we would very much like to work on our initial thought with some authentic data.

Speaking of the functionalities, we kept most of the features we had initially thought, albeit in an altered form, except a payment framework that needs some more involved learning of connecting our website with payment sites like Paypal, etc. which may not be considered as a learning objective of the course. One of our advanced database programs, i.e. a trigger to waive off delivery fee is a very sensible feature, which is unavoidable for any e-commerce website now-a-days, since the customers don’t and shouldn’t pay for a delivery fee regardless of their order size. To simplify the ordering process, we created a stored procedure that would automatically checkout an order and handle the inventory impact so that other users would not be able to purchase products that were out of stock.

A technical challenge that one of our teammates, Shantanu, faced was working with JavaScript to write the backend with relatively very little experience working with the language. This was particularly felt in the step where he had to create a login sort of feature, i.e. only entering the correct password leads to creation of an order. However, to get around the problem, he used SQL code, which feels even better, that he leveraged the learning in the course to solve a problem projected by something not taught in the course.

I (Elia) faced a similar problem. Although I had some experience building out sql integration in python, I have never built web apps/used javascript, so a big challenge for me was learning the relevant frameworks to create REST API endpoints for SQL queries and understanding how the frontend & backend worked together to create segmented methods of task responsibility.

I (Manish) had a problem while setting up our react page. Some of the functionality and the libraries which were compatible with version 5 of react had changed or had other libraries when I needed to set up our project. After doing multiple iterations and going through stackoverflow, I had to downgrade my react version from V6 to V5. There were many instances while creating our webpage that I needed to make sure that the packages which were installed were compatible with the V5 version of react.

While most of the application remained the same throughout our entire journey, some aspects as highlighted above did change. And as mentioned above, we would like to incorporate our personalized review feature using NLP for future work. Also, a limitation of query languages like SQL is the requirement of atomicity of the column entries, to conform with the first normal form. This prevented us from showing multiple products in an order as a list. We would like to explore NoSQL languages like MongoDB, etc. for this purpose.

Our team had very diverse members with expertise in different fields. This variety in skill sets was harnessed to divide the components of the project appropriately among the members. However, it did not prevent anyone from going out of their way to learn something new and use it to contribute towards the shared goal. All the members chipped in with their contributions which was very balanced in terms of division of work. Rather than assign specific tasks, each member contributed incrementally when they were available to the next piece of work required to complete the stages.