**Deliverable 4: Reflection**

Generally, we thought that process wise it went quite well. We started working on it chronologically, starting out from creating the entity relationship diagram to the SQL Query. We started out slow in the beginning because we had to complete the entity relationship diagram solidly to be able to create and alter tables in the database. However, we found out later that we had to alter the inserted data to be able to execute the outer join query so we went back to our SQL Database to add and update the inserted data so that it can accommodate the query. Not only that, we realized that order matters. We’ve encountered some problems because data in the ‘Many’ relationship table that has a foreign key could not be inserted to a table, and it’s because it should come later on. What we could improve on is organization in terms of writing the code, making sure that we are not skipping from one step to another and actually do it in an order that works.

Our project was sufficiently scoped because we strongly feel that we had incorporated most of the necessary tables that Cheesecake Factory would have. Since we already had 10 tables, we made some assumptions in order to simplify the entity relationship database. The process of inserting the data was prone to some errors because we had no real data to input and had to make up most of it. For example, we made sure that the dishes were actual dishes that CheeseCake Factory serves, and this was done just by picking random dishes from their menu. We also incorporated the real addresses of existing Cheesecake Factory chains across the United States. However, data such as suppliers, amount of ingredients, etc. are all made up and there were only around 11 entries for each table and we both felt that these data are not enough.

Every time we encountered an error in the SQL database, we tried searching for the appropriate solution online. One of the most common problems we encountered is during the table creation stage, particularly the data types. For example, we initially put ZIP code as a smallint, but ZIP codes with leading numbers of 7 or 9 are outside the range of smallint. We had to change it to a big in. But we also found out later that the ZIP code has to be in varchar(5) and not a bigint because Visual Studio eliminates the leading zero of the number (e.g. 02494 becomes 2494), which is misleading data to the user because the ZIP code is incorrect. Hence, we had to make several changes before we get the results we wanted.

This project enables us to learn how to create an ERD, build a database, and executing the queries, particularly knowing what kind of queries to execute in order to get a certain result. In addition, we get a better understanding of how to build the appropriate database and the right query functions to acquire selected data that will be beneficial and valuable for decision making or idea generation.

We had difficulty in changing the data types of the field and inserting data because once we created the table, we only had two options to change it. Either we use the design view or other functions such as UPDATE or ALTER. Since we did not want to complicate the Visual Studio database codes, we decided it best to add more data and using design view to alter the data types temporarily. We made sure to keep track of the changes we made in the design view. After, we finalized and reviewed the project by changing the data types in the function itself, deleting the CheeseCake Factory Database from the MSSQLLocalDB and re-run the whole code in our JBTS\_dbcreation.sql file and JBTS\_dbquery.sql file to make sure everything worked. Hence, it would be better to know the most efficient and easiest way to make changes so that we don’t have to waste a lot of time re executing and rerunning the entire thing.

Our team consists of only two members, so it was easier for us to communicate and divide the work. Since there are 4 deliverables, we initially planned to divide it into two deliverables for each person. However, we ended up working together most of the time since we encountered some problems and it was easier for us to discuss it together. For deliverable 1, we made the ERD together on a whiteboard before converting it to a Word document as we had to agree on the company and the tables. We also worked together physically on deliverable 2 and 3 because it was a lot of work for just one person and it can’t be shared on Google Docs, which was quite bothersome. We had to be in constant communication to see the changes made on the Sql and update each other what was wrong. Even using the GitHub was not as efficient as Google Docs. For the last deliverable, which is this reflection, we decided to work on Google Docs and just contribute as much as we can.

An issue that arises is that sometimes we encounter problems that we both can’t solve and we wasted quite a lot of time trying to figure it out. However, we think that overall we both worked together well and help each other in every way that we can. We both take proactive approach when encountering a problem by trying to find the solutions online or asking the professor. What we would do differently next time is probably coming up with a more efficient system of doing SQL as a group.