Final Team Report

Behavioral expectations:

Aaryan	Responsive	All Team members must respond to texts to make sure there are no communication errors.

These 2 expectations were met by both team members. Each team member remained vigilant in responding to texts which kept the other team member informed about what was being done. This helped reduce redundancy since it prevented members from working on the same thing their partner was. If any issue, unexpected bump, or outside problem came up, we kept each other informed about what was happening few days before we had to deal with those issues. This helped ensure that we were never caught off guard on our work as the other member could increase their workload to compensate.

Skill Identification

Tom	Research	If a topic needs to be researched, I will thoroughly learn about it. This is specially useful when we need to learn
Aaryan and Tom	SQL	Our internship over the last summer was SQL heavy so we are proficient at SQL hence can use it in several different ways if required.

Although both these skills were crucial, we did not delegate our work this way. Both partners worked on both aspects of the project depending on who had the time and how much work we had to do. Because both partners were proficient in both these skills, we worked on these aspects interchangeably. Research was done separately then brought together and discussed. Similarly, different parts of the SQL code was written by both members to speed up the process.

Participation:

Both partners participated on most aspects of the project. We met regularly to delegate work and check if we were on track.

Communication:

We talked more than once a week which allowed us to stay on top of each other's schedule and work.

Meetings and conduct:

We met at least weekly, and in our meetings we worked with each other with respect and let each other share our ideas.

Conflict:

We did not have any substantial conflicts during this project. The most conflict we had was deciding on which dataset to choose, but we ended up deciding to generate our own database since no database sufficient enough could be found.

Development process:

Research:

Both partners researched for the right dataset. Before the first submission, we decided that we wanted to work on social media dataset. Then we had to identify a dataset which we were unable to find because no data was large enough and diverse enough. Therefore, we decided to generate the dataset using faker.

Generating Data:

Data generation was done by Thomas, with some contribution from Aaryan, using Alqchemy and built-in Random libraries in Python. Approximately 1.5 million entries, or 3.7 GB of data, were generated for each table.

Queries:

Queries were written by Aaryan, with some contribution from Thomas. While the original queries were written using SQL, they had to be moved to Python SQL queries because Kaggle is the only software large enough for our purposes and it does not allow direct use of sql.

Insertion and Rollbacks:

Insertion and Rollbacks were made by both partners. These queries were also made using SQL, but had to be moved to python SQL due to the data size requirements.

Reflection:

This development process allowed both partners to utilize their skills extensively and further develop our knowledge of SQL. Both partners learned crucial new skills including how to generate data, and crucially how to use Python to write SQL queries to harness the qualities of both languages. We also learned how to connect our queries to other software at the backhand, which helps improve data querying. These skills are crucial for database management in the future.