SQL Reflective Piece:
In this document I will be briefly discussing the assumptions made and the places for improvement in our SQL group project.

The most obvious flaw with our statements is the lack of futureproofing. In the context of the assignment we had to make the assumption that the database was not going to significantly grow or change anytime in the future. The addition of an extra column or table would most likely break our code and would require manual fixing. Also, our statements are poorly optimised and if the database grew significantly there would most likely be longer waits for statements to be executed. One thing to minimise this issue could have been implementing the LIMIT keyword to reduce the amount of data shown to the user at once.

There is also space for improvement in the security of our database management, customer's personal details should not be so easily accessible and this is important for a company operating on a commercial scale. Ideally within a large company, people should only have access to the parts of the database that they need to know to do their job. Our statements implement no safety features and therefore would be vulnerable to a security breach.
There is also a minor issue with our use of the wildcard operator. Selecting everything from a table without any filtering or vetting could lead to security issues and at the very least can be inefficient on a large database.

Because of my relative inexperience with SQL, I feel like in some of my statements there are ways they could have been more concise and less long winded. This would have improved the readability and the versatility of my code. Take for example my answer for question 13: "SELECT invoiceId, customerId, Total - (SELECT AVG(Total) FROM 'invoices') AS priceDiff FROM 'invoices' WHERE Total > (SELECT AVG(Total) FROM 'invoices') AND Total - (SELECT AVG(Total) FROM 'invoices') > 15". Even with my use of aliases it is difficult to tell what the statement is actually selecting for. This is because in the context of this assessment we were primarily focused on just getting the code to pass the tests, I know that no one is going to have to work on this code years down the line so future readability was not a priority.