## Feasibility Report

### Introduction

This project proposes the creation of an application that provides a GUI for modifying and viewing SQL databases, similar to Excel by handling the syntax by allowing the user to input their queries through buttons and fields. The intent to provide a more ergonomic and faster way to and less error-prone database handling.

### Background

Modern commercial and consumer-grade computers are powerful machines capable of incredibly complex tasks, but unless you are performing actions by writing in pure machine code and even then, there is automation happening within your machine to allow these actions to be performed. From operating system’s built-in operations to the IDEs and compilers and the software built from them, increased layers of automation can allow even someone with basic computer operating literacy to perform a wide and complex level of tasks from writing documents, browsing the internet, and producing art. As these increasing number of platforms introducing new layers of automation are produced, even higher levels of automation can be adapted specifically for these functions to allow a wide range of operation to be performed quickly and more availably. For example, as relevant to my project, SQL, as well-known database language was built upon C language (Cunningham, date unknown), and this can be taken a stage further (further than the IDEs, that is) and create a more intrinsic and automated application.

## Outline Project

My proposed project attempts to create a (albeit in a small scale) bridge between a more technical knowledge-required operation of databases to a more intuitive approach where syntax and format is handled automatically to allow a user to create, view and modify a database with ease, similar to Microsoft’s excel. Not only does this allow those who have not learnt the more technical operations of database languages to perform these tasks, but also a more automatic approach for those who are already fluent. Therefore, while making the features SQL is capable of in a more readily available format, I will intend to present similar terminology and structure that the language presents in a way that is more inherently comprehensible. It can be considered a GUI program for using an SQL database.

One possible further approach is to create a basic database application without SQL but rather by introducing a new database frame produced and made readable only by the program itself. This would however create an exclusivity which I would like to avoid for it would be re-iterating ground that has already been covered with much more sophistication than the scope of this project can encompass and with no further benefit operationally other than possible ease of user comprehension though this can be achieved through how the GUI is implemented anyway.

### Conclusion

This project, in its current proposal provides a fairly wide scope allowing for a variety of solutions towards its intent that can be defined from further design and requirement development. Even in a fairly basic operational form, it can achieve its objective quite easily if designed carefully. While there a variety of complexities that can be implemented, the primary focus of ease of use built upon using databases must be held as the focus.

### References

Cunningham, L. (date unknown) PL/SQL features by release. Available at: [PL/SQL features by release (dba-oracle.com)](http://www.dba-oracle.com/t_edb_pl_sql_features_release.htm)