The features contained in the jupyter notebook file are exactly to project specifications. However, there are limitations when performing the insert, update, and delete of records from the database. For insertions, the user is required to provide all fields of information for a given record. In the case the user does not have all the information on hand, the substitution of dummy values is highly recommended by the developer. Given the extremely varied nature of the criteria on which tuples can be selected for update/deletion, a single (usually the primary) key is used to locate these tuples. As such it is necessary for the user to keep in mind the ID numbers of the records they wish to affect. This can have unintended effects when updating or deleting the tuples from Views, and so the statement should be used with care.

Now, given this information, there is a substantial amount of area in which this application could be improved. A more user-friendly UI could assist in developing a more robust query-generation system, and is decidedly implementable all within jupyter notebook using python widgets. This would result in an easy-to-use method of dealing with the staggering amount of variety in potential user queries and make it more accessible to these users. With an even greater time commitment, it could be possible to give the users a detailed graphical view of the schema of the database complete with the tuples already extant in each table. This would enable the user to quickly find the information they need and construct the queries they require. Of course, all these methods would require greater attention paid to the transactions and indexes created for this particular database. Since the test product is rather rudimentary, not much time was spent on creating a proper indexing system to greater increase query efficiency, given the limited scope of records to be handled. With all these in place, the application would be rendered both more convenient for end-users, and more efficient in its interactions with the database server.