Database Evaluation:

1) Identify the purpose of the database and the purpose of each table.

The purpose of this database is to maintain records relating to drug prescription to track what drug was prescribed, who prescribed the drug, and who it was prescribed for, as well as relevant details for each. The purpose of the *prescription* table is to maintain a record for every prescription that is written, including the date, pharmacy, physician, and patient involved. The purpose of the *drug* table is to keep records of each of the drugs used at these pharmacies. The *pres_drug* table is used to keep track of which drug was prescribed for a prescription as well as dosage and dates. There are also the *pharmacy* and *physician* tables which serve to maintain distinct records of all of the pharmacies whose information is being stored here as well as a list of all of the physicians who have written prescriptions that are fulfilled at these pharmacies. The *office* table keeps records of each physician's office. The *patient* table tracks all of the patients involved at these pharmacies and their relationship to prescriptions. Finally, the purpose of the *consultation* table is to record when consultants are utilized and their relationships to the patients and doctors.

2) Describe the relationships between the tables, and why a particular cardinality (1-to-1, 1-to-many, many-to-many) was chosen.

The relationship between the *pharmacy* and *prescription* tables is one to many, since for each pharmacy record there may be many instances of its *pharmacy* ID in the *prescriptions* table. This makes sense since one pharmacy can have many prescriptions. The relationship between the *prescription* and *pres_drug* tables is also one to many since there can be many different prescription ID's in the *pres_drug* table but a certain *prescription* ID can only appear once in the *prescription* table. The *pres_drug* table to *drug* table relationship is many to one since there are many possible *drug* ID's in the *pres_drug* table but in the *drug* table each drug ID is unique. In the same way, *Physician* ID's and *Patient* ID's can occur multiple times in the *prescription* table but only once in their parent tables, *Physician* and *Patient* respectively, where they are unique. In other words, for one physician there can be multiple different prescriptions. The *Office* table contains records with unique *office* ID's, but in the *physician* table there are many possible office ID's, so the relationship in this case is one to many *Office* to *physician*. The *physician* and

patient tables are also related one to many since many possible *physician* ID's could occur in the *patient* table, but they are unique in the *physician* table. Finally, in the *consultation* table there is a many to one relationship between the *patient* ID and the *patient* table and the *physician* ID and *physician* table because in each instance there are many possible *patient* and *physician* ID's that could appear in the *consultation* table, but in their parent tables there will only be one unique record. Whenever a primary key is part of a relationship it is always one to many since primary key's are ID's for unique records but when they are foreign keys they can occur multiple times. Also, each different entity involved in the process has its own table with details, such as a table with just patients. This prevents the storing of redundant data in multiple tables.

3) Identify any issues that you could find in this database.

The *consultation* table contains the patient and physician involved, but it does not relate to the *prescription* table. I think this is an issue because it provides a set of records that are only useful if you want to see which physicians have worked with which consultants. It also does not contain supplementary details about the consultation such as if it confirmed a prescription or had no effect on a prescription. To fix this problem, I think the *consultation* table should include the *prescription* ID as a field. That way a database user could determine what prescriptions were affected by consultations. It might also be useful to add a few more descriptive fields for the *consultation* table.

4) How would you improve the above database?

In addition to the steps outlined in question 3, I would improve the database by adjusting the *pres_drug* and *prescription* tables. I think that the dosage, start date, end date, and refills fit in more with the context of the *prescription* table so I would move those fields there. That leaves three different ID's in the *pres_drug* table. Essentially these just relate the *prescription* and *drug* tables, so I would cut out the *pres_drug* table all together and simply include a *drug* ID in the *prescription* table. The same amount of information would be stored and it would remove an unnecessary link between the prescription itself and the drug prescribed.