Katie Craven

Lab II

2) A super key is a set of one or more columns that can identify a row. It can contain extraneous columns not necessary to uniquely identify a row. A candidate key is a minimal set of columns that identify the row. A primary key is the key chosen from the possible candidate keys to uniquely identify rows in a database.

3) One might create a table for my friends. They could have columns for first name, last name, nickname, middle name, age, gender, and friend ID (FID). First name and last name are both text datatypes. They should not be nullable in that everyone has a first and last name, and therefore there should be no reason for them to be null. If I did not know my friend's first or last name, I could put in null rather than incorrect data, but it would be problematic. Middle name is a text datatype, and it is nullable, as not everyone has a middle name. Age is numeric (an integer, more specifically) and should probably not be listed as nullable, but if I do not know the age, it could technically be nullable. Gender is text, and should not be nullable, as everyone has a gender. Friend ID is a numeric (integer) data type that is a unique identifier for each friend, and is assigned as they are added to my friend database. This field type cannot ever be null.

4)

a) The first normal form rule is that each attribute contains only a single value. This is important as it would mess up the kinds of relationship things have to each other. If you are searching for a single element and receive a list of elements, or vice versa, it could break the search.

b) The access rows by content only rule ensures that your data is being accessed properly. It would be incorrect to ask for the third name in table "friends" because that organization may change and will give inconsistent results. You could instead ask for the friend with FID 3, for example.

c) The "all rows must be unique" rule is important because duplicity in databases takes up space and can lead to confusing results. Say there is an Arthur T. Daniels and an Arthur D. Daniels. These are put into a database where only the first and last name are recorded. You now have two different people, but according to the database they are the same. There lies the potential for these two Arthurs to get confused in the database, as well as the potential for a search for just one Arthur to return two Arthurs.