Database Design:

Looking at the requirements of a quiz program for a number of users, it can be seen that an amount of data persistence is required, for different sections of the program. The two main objects requiring persisting data are the users, and the quizzes themselves.

While it is technically possible and initially easier, to code in a set of values for the quizzes, for the number of words that would ideally be required, as well as to have them easily editable, this is not an acceptable solution. As such, a table within a database dedicated to storing word data is required, not actually interacting in any way with any other tables.

# Users:

To support any form of individualised functionality in a system, a login system of some form is required. As such, each user requires a username and password to log into the system. To support different user types, some form of value must also be stored in relation to each user, to store their user type, or access level.

For any kind of permanence from the quizzes, records of the results would need to be stored. Therefore a record of a quiz a user has taken, will be stored, in a results table, with a link to the user that took that quiz, so only they (and possibly admin users) can view it. Since different types of quiz are available, that too will have to be recorded.

## Login Table:

userID: identifier for this particular user

Username: the human readable identifier a user uses to log into the system

Password: the password a user uses to log into the account. Ideally this would be a hash instead of plaintext, using a slow hashing algorithm, as well as a salt. A third party, token based login system, like google’s, could also be used.

userType: flag to identify the type of user: student, instructor, or administrator.

## Results Table:

quizID: identifier for this particular quiz session

UserID: ID of the user that took this quiz

QuizType: the type of quiz that was taken

Result: the total number of correct answers given by the user

OutOf: the number of questions the quiz was out of (the values of score could have be stored as just a single value, or as a percentage. However, with this implementation, this allows the number of questions in a quiz, to change, without having to recalculate any values, and also potentially allows for a wider variety of metric tracking in the future).

DateTaken: the date that the quiz was taken

# Quiz:

## Word Table:

An important distinguishing point to realise early on, is that working from my native language (english) outwards to welsh, in the design of how to store words for the quiz to call upon, would be a rather short sighted choice of action.

English nouns (since this is all we are working with initially, but also a convenient example case) are usually spelt the same, and are the same “word”, irrespective of the tense there are being referenced in, whose possession they are in, what they are in relation to the speaker, or any other circumstance of the noun. Furthermore, a number of languages classify all objects as male or female. English does not do this, so a “Teacher” could be male or female, and there is no way to know from the word itself. In other languages, the spelling of the word changes depending on if the, teacher in this case, is male, female, or neutral (referring to the word teacher in a general case, opposed to a “case of a teacher”).

Therefore working from welsh words, backwards to english, seems like the sensible option, as in most cases welsh appears to have a finer granulation of vocabulary than english.

In any case, with my method of storing words for the quiz to draw from, it should be possible to support words having a gender (or no gender at all), and therefore repeat english definitions of words. Support is also be given to be able to identify the type of word, a word is, for possible expansion of the program beyond just nouns.

Therefore words have the following attributes within the database:

WordID: to identify each word entry

Welsh: the welsh word from which the rest of the entry stems

English: closest english single word translation of the source welsh word

Word Type: a flag to determine the type of word, this entry is (noun, adjective, pronoun, etc…).

Word Gender: the gender of the word according to the origin welsh word / spelling. Can be male, female or neutral.