**College Records (and) Programmes Database (CRPdb)**

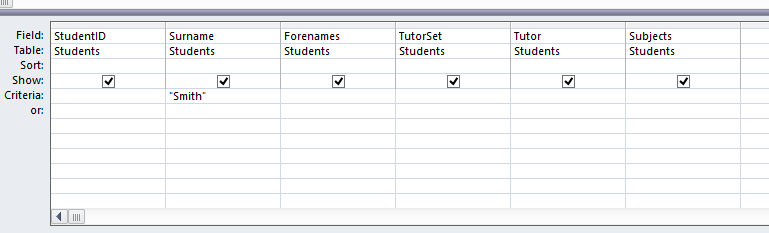
This exercise is designed to illustrate some of the problems associated with database design. The CRPdb Database contains a ‘flat file’ structure, hence its name!

First download from Godalming Online the file “CRPdb.zip”, save it and unzip it. The file can then be opened with MS Access.

If you are working at home, you can either try to download a copy of Access from your college office 365 account, or you can try to download the MySQL version of the database and run it through your Xampp server. Many of the features described below are for Access, but you should be able to do ‘query by example’ using the search tab in phpMyAdmin.

The database contains one table called Students (ignore for the moment the table Students2, which is a failed ‘attempt’ to improve on the first version).

In query design view, write a ‘query by example’ as follows to select students with the surname Smith. There is no need to type the quotes, these get added automatically.



Switch to view (left hand button on query ribbon). How many students are selected and who are they?

………………………………………………………………………………………………………………………………………………………………………………

Now modify your query to look for students with the forename Thomas. Does it produce the desired results? Look at the table and see if you can explain why.

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

Try instead using a ‘wildcard’ character, \* (or % in MySQL), which will match with anything. Type \* Thomas (no spaces or quotes) in the criteria box. Is a student selected this time and which one?

………………………………………………………………………………………………………………………………………………………………………………

Also try using Thomas \*. What happens this time?

………………………………………………………………………………………………………………………………………………………………………………

Can you manage to modify the query so that it selects all students with Thomas as one of their forenames?

Now we will concentrate on the subject data. The problem here is very similar. We want to know which students study Physics. What would you expect to happen if you typed Physics in the Subjects criteria box?

………………………………………………………………………………………………………………………………………………………………………………

Try this and see if your prediction is correct. Now, once again, see if you can modify your query to select all students who study Physics. This should be possible, but it is not very elegant and points to why this flat file structure is not ideal.

The table Students2 is an attempt to improve the situation by allocating each subject to a separate field. Take a look at this table and then write a query to select those students who have Physics as Subject1. Does this select all students who study Physics? Why not?

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

The answer to these questions should be fairly obvious – we still basically have a repeating attribute (subject), but in a different format. Try to modify your query to select all students who do Physics. Again, it should be possible, but is not the optimum way to do things.

There is a further problem with the use of 3 separate subject fields. Brian Boffin is very frustrated by this structure – can you explain why?

………………………………………………………………………………………………………………………………………………………………………………

Now revert to the original table and query and modify the query to select Michael Lancaster (the match can be done on the surname). Michael’s Personal Tutor has changed from PJM to JMH. It is possible to modify data in the query view grid. Change the tutor and revert to design view – there is no need to save, as Access saves all data changes automatically. Now modify the query to select a list of students in tutor set 7F. A problem should immediately be apparent! What is it?

………………………………………………………………………………………………………………………………………………………………………………

The correct tutor for 7F is now JMH. Modify the output data so that it is correct. This may not be too much of a problem for you, but would have been more of a headache with a typical tutor group of 20 or more students. In general, what problem with the data structure does this illustrate?

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

Now write a query to select Brian Boffin and change his tutor group from 8D to 7F. Once again, select a list of all students in 7D with all of their data displayed. A similar problem should now be evident with Brian Boffin’s tutor. What is this?

………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………

Of course, it may be that you took remedial action to prevent this problem from occurring, but a well-designed relational database structure should prevent problems like this from occurring in the first place. If you make a change to an item of data, it should be reflected in all related records in the database.

We are basically talking about issues to do with data consistency and redundancy of data storage. If you can understand this, then you are a long way on the way towards understanding database normalisation.