

CS200 Assignment 9 – FileMaker

Due Monday July 20, 2020, 11:59 pm

Readings and Resources

On Learn: *Database Resources > Database Design*

Files

The following are located in Assignment 9 Files on Learn.

- Favourite Food:** This is a text file to import into the database you create in question 1
- FavouriteFoodDemo:** This file shows you how your question 1 will look and work in the end.
- University Starter:** You will use this file for question 2 of the assignment.
- UniversityDemo:** This file shows you how your question 2 will look and work in the end.
-

Assignment Objectives

- The idea of a “self-join,” (and how you write a self-join in SQL), which in FileMaker is a relationship in which a single table is linked to itself.
 - An understanding of how to implement (some) three-table joins in FileMaker.
 - Be aware, however, that more complicated select statements than we have considered can be far harder to work out than the simple cases we have considered.
 - The notion of a global field in FileMaker. In many respects, global fields serve as variables in FileMaker. As we’ll see when we discuss scripting in FileMaker, you can create a layout with which users give global fields values, with buttons that execute scripts that use those values while manipulating data.
 - The notion of creating a value list from the values stored in a given field of some table, and requiring the user to select from that list while entering a value.
-

Assignment Strategy

This assignment may be done independently or with a partner. If you do work with a partner, please add both or your names, usernames and Student ID numbers to the footer of each layout in the files and name the files with both usernames (username1_username2).

Your Tasks

1. [50%] Sometimes it is useful to form the join of a table with itself (self-join). Consider a simple table (shown on the right) that records each person's favourite Food. (Because a person's name is not necessarily unique, there is also a unique sequence number called pKey for each record.)

FavouriteFood
pKey
Name
Food

- a) Create a FileMaker table named username_FavouriteFood with an integer primary key field and two text fields called Name and Food.
 - In defining the table's fields, specify that all three fields must have a value, that the primary key be unique, and that a
 - sequence of serial numbers be automatically assigned ("auto-entered") to pKey (as shown above left).
 - Import the data contained in the file FavouriteFood.txt, which you can find on **Learn**.

Serial number

Generate: On creation On commit

next value increment by

- b) Create a data entry layout called "Input" (shown to the right) that allows you to enter new records. For the Food field, users should be forced to select a person's favourite Food from a popup menu of the Foods already entered for this field in another record. You do not need to type the list of Foods. Also allow users to enter a previously unused Food.

- c) Create a layout called "MyFriends" showing the current record's fields and a 10-line scrolling portal listing the names of all those records having the same favourite Food as that of the current record.

- To create this portal you will have to define a relationship from FavouriteFood to FavouriteFood in which the Food fields are required to be identical.
- Name the duplicate table "FindSameFavouriteFood".
- Since you don't, however, want to list yourself as a friend, you will also need to require that matching records not have the same value of pKey.¹
- Prevent users from editing ANY of the fields displayed in this layout, including related fields within the portal.

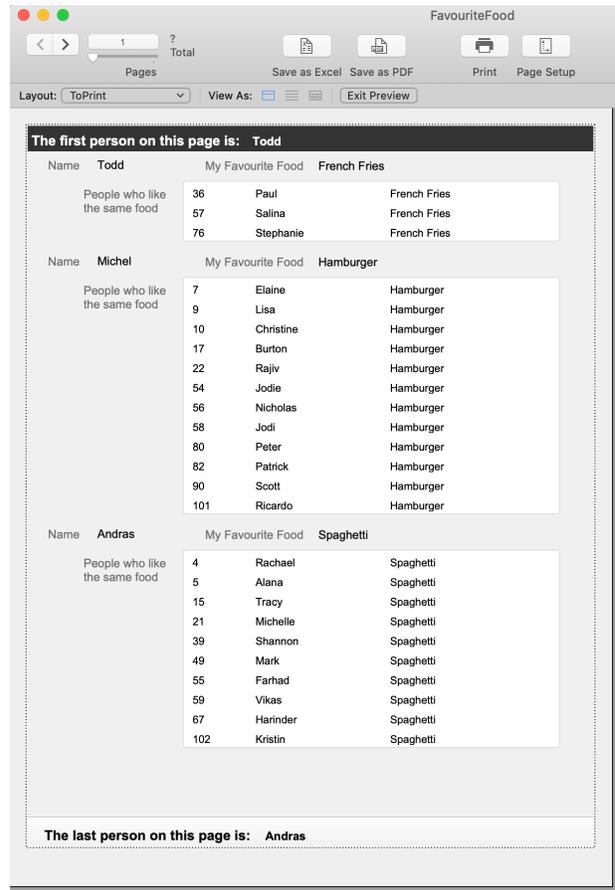
¹ The SQL Select statement for this is

```
select A.pKey, A.name, A.Food, B.pKey, B.name, B.Food
from FavouriteFood as A, FavouriteFood as B
where (A.Food = B.Food) and (A.pKey <> B.pKey)
```

- Add a fourth field having the name “HowManyFriends” to FavouriteFood and display it on the MyFriends layout. HowManyFriends should be a calculated field; its computation should be `Count(FindSameFavouriteFood::pKey)` ².
 - Your MyFriends layout should look like the example shown to the right. (Please put your name(s) and username(s) in the header.)
- d) Create a layout called “ToPrint” (like that appearing below). This layout will show the names of others having the same favourite Food as the person listed on the left.³ The thing to particularly notice here is that the list of people is of variable length. This is accomplished by creating a portal that has more rows than you ever expect to actually show (try counting the records and make an estimation).

- Use the inspector’s “Position → Sliding & Invisibility” to instruct FileMaker that the portal (and the part containing it) should collapse vertically (“slide up”) to eliminate unneeded space.
- The header and footer of ToPrint should contain the name of the first and last people whose records appear on the page (as shown above and in the demo).

Please put your name(s) and username(s) in the header of all layouts, and the page number in the footer (refer to FileMaker’s online help, or the Poke Pearl for inserting special variables into layouts).



²The SQL Select statements for this look like

```
select count(*)
from FavouriteFood as A, FavouriteFood as B
where A.pKey = 15 and (A.Food = B.Food) and (A.pKey≠B.pKey)
```

2. [50%] This question has you create a 3-table join of the Students, Marks and Assignments tables you will find in the University Starter file. These are slightly simplified versions of the tables by the same name that you met in the SQL assignment. They contain the following fields (fields comprising the primary key for each table are shown in bold):

<i>Students</i>	<i>Marks</i>	<i>Assignments</i>	<i>Register</i>
IDnumber	IDnumber	Course	ID Number
Surname	Course	AssignNum	Course
Initials	AssignNum	Weight	Section
Street	Mark	Description	
City			
Province			
BirthDate			
Gender			

- Download the file UniversityStarter.fmp12 from **Learn**. Then create a layout in the **Students** table that looks like the following and behaves like the sample solution to this question.

Read all instructions before starting.

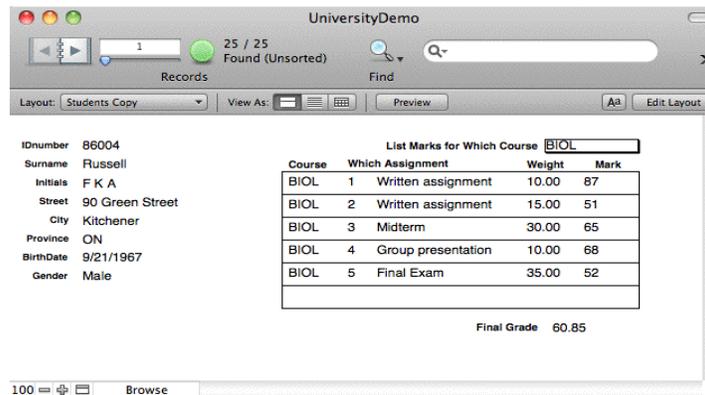
For a given student, the portal lists the marks for the student in the class selected by the List Marks for Which Course? popup in the upper right corner. It also shows the student's mark in that course, which is computed by *multiplying the student's mark for each assignment times the weight for that assignment*.³

There are lots of interesting things about this layout.

- How do we specify which course we're interested in? Which is to say, what is the List Marks for Which Course? popup connected to?
 - One possibility is to create an auxiliary text field (call it WhichCourse), and require the user to enter a value into it by selecting from a popup menu connected to a value list whose values are taken from the Course field of the Assignments table.

³ This layout implements questions 1(g) and 1(h) in the SQL Assignment, though more generally.

- That would work, but would require allocating extra space for the field in every record, which is unnecessary.



- Instead, WhichCourse should be created as a “Global” field. These are “pseudo-fields” whose values are shared by all records. Only one value is stored in the table, separate from the records in a table, and that value is seen and shared by all records in the table. You can still format this as a pop-up. (Be sure to declare that WhichCourse is of type Text, then click the Options button for WhichCourse and find the checkbox that makes it a (shared) Global field..) It would be easy to list ALL the marks for a student in the portal – you’d just define a relationship using the IDnumber field as the link. But we want to show in the portal records from the **Marks** table for a particular IDnumber *and* Course, so you want to match the values in *two* fields as you create a one-to-many relationship from Students to Marks.
- The course, assignment number, and mark columns of the portal are just fields from the Marks table, which are readily accessible once you have connected the Students and Marks table. Place these fields into your portal.
- What about the portal columns containing a textual description of each assignment and the assignment weight? Those textual descriptions are stored in the Assignments table.
 - It seems we need to connect the Marks table to the Assignments table. What is the primary key for the Assignments table? The Course *and* AssignNum fields. Once again you need to link two tables by matching the values of *two* fields.
- Finally, the course grade shown bottom right. See if you can figure out how that’s done. You’ll need to create a calculated field in the Marks table, and a calculated field in the Students table, but nothing more. Details left to the reader...

Password-protected solutions to these questions are in Learn. (Use a blank password.)

Bonus

[10 marks] The following are some examples of what you can do in SQL's WHERE clause:

- **compare a field value with a literal value**
example: name = 'Tony'
- **compare two field values —often used in a relationship condition**
example: Students.IDN = Register.IDN
- **combine comparisons using an OR**
example: (Course = 'Biol') OR (Course = 'Phys')
- **combine comparisons using an AND**
example: (Surname = 'Tony') AND (Students.IDN = Marks.IDN) AND (Course = 'Biol')
- **other assorted operators can be used**
example: (Mark > 80) AND (Mark < 90) AND NOT (Studnum = 8912345)

The above are different examples of what you can do with an SQL where clause as opposed to a FileMaker Find. **Explain how to do each of the above queries in both SQL and in FileMaker's Find Mode** (or, when necessary, explain how to do the query in one application and note that it is not possible to do in the other). **Note: This is essentially a compare and contrast question.**

This question is not asking how you would do the particular SQL example given in FileMaker, but how you would do the bolded statements in both applications (i.e. do not explain how to find "Tony" specifically).

Assume that your reader is an experienced user of both SQL and FileMaker.

Please enter your answer in a properly (but not elaborately) formatted MS Word document named **username_a9bonus** (or username1_username2_a9bonus) of at most one page showing your name and username (or codes, if you are working with a partner) at the beginning of your document.

Comment: this is the sort of question you should habitually ask yourself when you have available two applications, either of which might be suitable for the same task. It is also a typical CS 200 examination question, for which this is good practice.

Submission Instructions

- Create a folder called username_Assign9
- Move each of the assignments files (username_A9Q1 or username1_username2_A9Q1, username_A9Q2 or username1_username2_A9Q2 and username_a9bonus or username1_username2_a9bonus) to this folder.
- Compress this folder and name it username_Assign9.zip and submit it to the **Assignment 9 Dropbox** on **Learn**.