WORKSHEET

DATABASES INTRODUCTION

WHAT IS SQLITE

SQLite is free database software that you can download from sqlbrowser.org SQLite is a great piece of software to learn databases and to create small local databases for your projects.

This section looks at setting up a database using SQLite. The E-sports theme is continued from the last section.

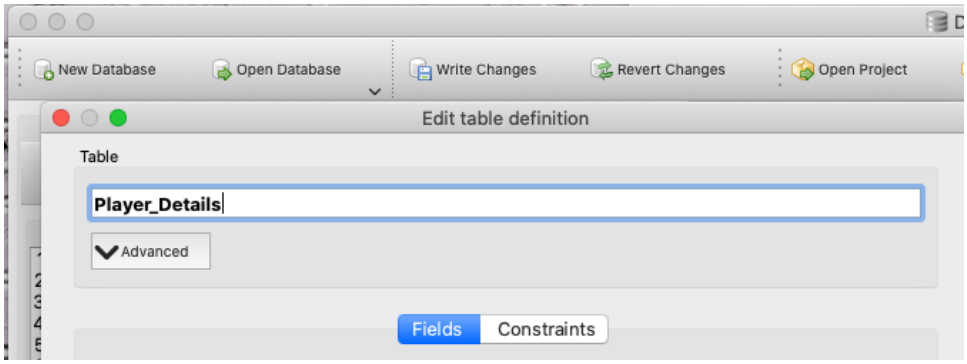
The next sections break down creating a database in SQLite into 4 easy steps:

1 | Create a new Database
2 | Add fields to tables
3 | Add data to each table
4 | Produce a query using SQL

A | CREATE A NEW DATABASE

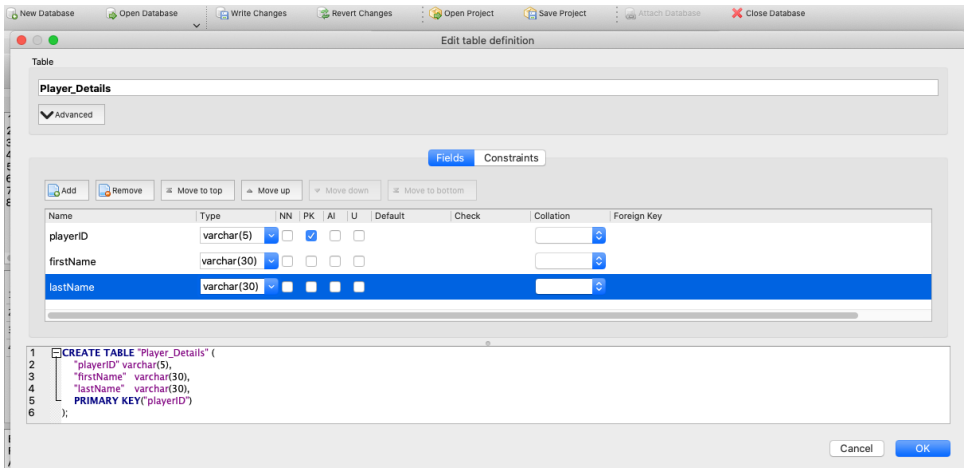
After downloading SQLite from sqlitebrowser.org install the software using the normal methods from your operating system. The SQLite icon has three grey disc like the image shown.

i | Open the software and click New Database, you will the need to give a meaningful file name and try to avoid and special charters.
ii | After creating the file a window should pop-up ready for you to create your first table. Enter the name of your first table. If possible try to select one of your tables that does not have a Foreign Key for your first table, no problem if this is not possible.



B | ADD FIELDS TO THE TABLE

i | Click on the 'Add' button to add your first field
ii | Add the data type for the field. Note: you can manually type the data type, you are not restricted to the dropdown list, demonstrated below using varchar(5)
iii | Select any of the tick boxes needed (PK) Primary Key, (NN) Not Null, (AI) Auto Increment or (U) Unique.
IF you have a Foreign Key the next section will show you haw to do this.



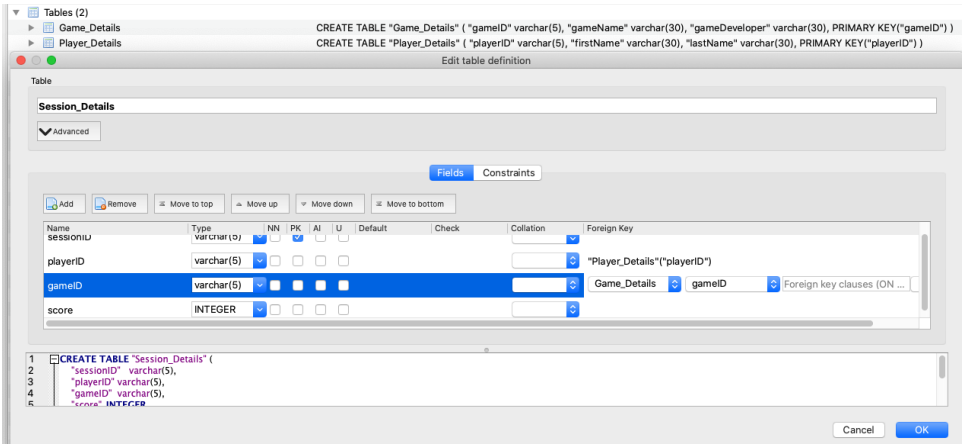
Remember to press Enter after adding the last field and **double check** it has been added to the SQL statement a the bottom of the window. Then click OK to confirm.

After you have added created your first table you are now ready to create the next. In this example a Foreign Key is added to the table to relate to the table already created.

i | For the next table repeat the previous instructions by clicking 'Create Table' in the 'Database Structure' section.

ii | To add the (FK) Foreign Key double click in the Foreign Key section in line with the field to add to FK, slightly confusing because the FK details do not show until you double click in what appears to be blank space.

iii | You can the assign the Foreign Key and assign which table the Foreign Key links to.



Remember to use your ERD (Entity Relationship Diagram) and ensure you get all of your field names and data types correct, otherwise the relationships will not work.

C | ADD DATA TO EACH TABLE

In the **'Browse Data'** section

i | Click the 'Insert New Record' icon to add new data. As highlighted in the image below.

ii | Add your data. Click the TAB key after adding each item of data to go to the next.

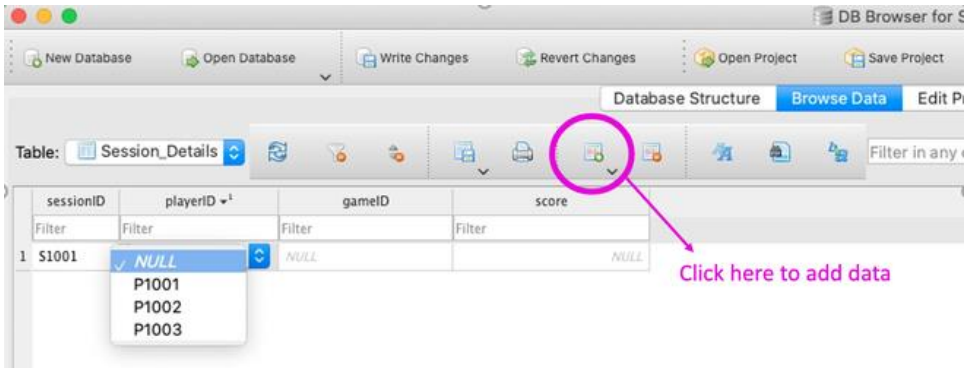


Image A

NOTE: You will need to consider which order you add data to each table, if data in one table relates to data in another then you will need to create the data in table that you will relate to before you try to relate to it. When done correct you should get drop down menus as seen in 'Image A', these are the links to related tables of which the data you have already populated.
Remember to save your database.

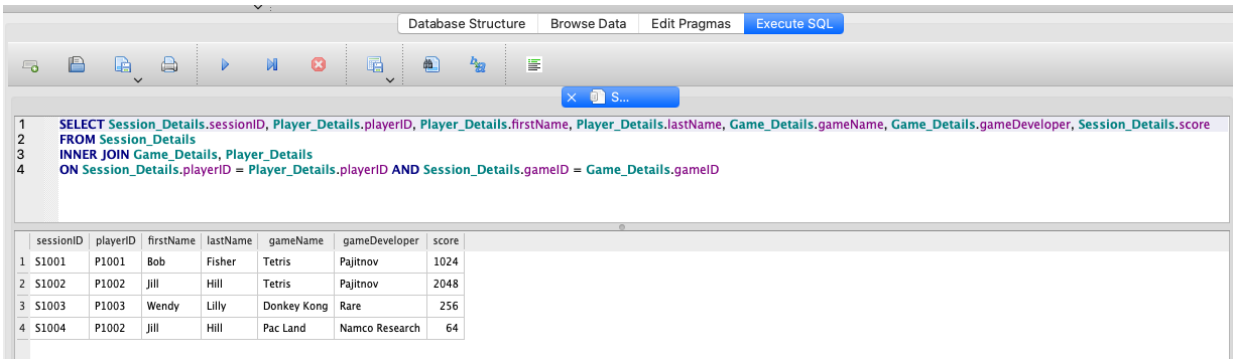
D |PRODUCE A QUERY USING SQL

In this section the query demonstrated will create the outcome as can be seen in 'Figure 1

	sessionID	playerID	firstName	lastName	gameName	gameDeveloper	score
1	S1001	P1001	Bob	Fisher	Tetris	Pajitnov	1024
2	S1002	P1002	Jill	Hill	Tetris	Pajitnov	2048
3	S1003	P1003	Wendy	Lilly	Donkey Kong	Rare	256
4	S1004	P1002	Jill	Hill	Pac Land	Namco Research	64

i | Click on the **Execute SQL** tab
ii | Choose the fields you want to include in your query with the **SELECT** command. You need to select the table and the field from the table. For example to get the session ID from the session table and the players first name from the player details table the command would be:
SELECT Session_Table.sessionID, Player_Table.firstName
iii | Choose which Table hold the links to the data you want to return by using the **FROM** command, in this case it is the Session_Details table.
FROM Session_Details
iv | Choose which tables that are linked to this table that you want to access details using the **INNER JOIN** command.
INNER JOIN Player_Details, Game_details
V | Choose which details you want to return using the **ON** command, for example in this case if the playerID is in the Session_Details table then return the Player details from the same player from the Player table. Remember stage ii dictates which of the details will be returned.
ON Session.PlayerID = player_Details.playerID

Click the **RUN** icon to check your SQL statement



SQL statement with outcome

As can be seen from the image above, all fields to display have been added in the SELECT command. The last line with the ON command has linked to the Player_Details and the Game_Details table using the Session_Details Table playerID (FK) with the Player_Details Table playerID (PK) and the same method for the Game_Details link.