

PHPM 631 Assignment 8: Writing SQL Queries
Midpoint (Part 1) - Due date: Submit on E-Campus by 11:59pm Sunday 4/21
Assignment 5 - Due date: Submit on E-Campus by 11:59pm Sunday 4/28

Submission. Submit on E-Campus. See Requirements Section Below for details

- Midpoint (part 1): Writing SQL Queries
 - Upload your own SQL Queries to answer the questions given (May work in groups)
- Assignment 8
 - Upload your SQL queries code (May work in groups)

Late Assignments: Due to the end of semester schedule, no late assignments are allowed for this assignment. Submit whatever you have done by the due date.

Plagiarism: If you consult any outside sources when doing your work, you are expected to further document these sources. Give credit where credit is due. Plagiarism will not be tolerated.

Guideline for assignment grading (8%)

- 70% (70 points): Bad (√ --) Did NOT follow all instructions
- 80% (80 points): Reasonable (√ -) Followed all instructions
- 90% (90 points): Good (√) Followed all instructions, and did good work
- 100% (100 points): Great (√ +) Followed all instructions and did great work

Objective

By the end of this assignment, you should be able to

- Write SQL queries to answer specific questions about the information in the database

Assignment 8: Writing SQL Queries

It's now time to look at data

You should have started to learn how to write SQL queries through w3schools.com for quiz 10. In this assignment you apply what you learned to your own SQL queries.

Writing SQL Queries

Now it is time for you to write your own queries. If you have learned properly for the quiz, this should be easy. If not, you should take the time to learn what you missed during the quiz.

You are encouraged to work in a team of 2 for this assignment. If you do, both of you will receive the same grade for this part of the assignment. It is IMPORTANT that BOTH of you submit on E-campus. Each of you are REQUIRED to type your own SQL query without copy/pasting. BUT at the very top, clearly label that you worked as a team and the name of the team members. You may choose to work along, but you will be expected to do the same amount of work as a two-person team.

YOU WILL HAVE TO WRITE SQL QUERIES FOR THE FINAL TAKE HOME EXAM.

In this assignment, you will be using SQLite3 (a variation of SQL) to write queries to answer questions given below using a database I provide. For this assignment, you will have to use a software called sqlite browser. It is installed in the lab, so you can do the assignment there. Or you can install the small software on your own computer, at the end of the assignment are general instructions for how to install on a windows based machine and a Mac.

How will you use SQLite3 (Recommended. See last section for another option)

1. Once launched click on "Open Database" and upload the "kum.db" database (from the class website).
2. To run sql queries click on "Execute SQL". Write your queries and click on forward button (▶) to run.
3. You can also run using F5 or Ctrl+enter.
4. Copy and paste your final SQL queries into a separate text file. Collect all SQL queries for your questions into one text file and submit with the answers to questions. This is what you submit.

5. You can save the project to various locations as well as download project to local machine. But you are not making any changes to the DB so there is no need to save anything. Personally, I find it is better to just use a “fresh” shell and repeat the following steps each time because it is simple. But, it might help for you to keep record of your SQL queries as you work on them in a separate text document so you don’t lose them. **Remember to upload kum.db to any new session.**

Recommended Action Plan

1. First, do step 1 above
2. See what tables you have in the database: click on the **Browse Data** tab
3. See what variables are in a table. This is often called the schema of the database (the tables, and the list of variables in the tables)
4. Look at all the tables, one table at a time. You should also try to review the tables using SQL queries. Remember your semicolon at the end. (HINT: **select * from table_name;**)
5. Now you should know what is in the database.
6. It’s time for you to start the assignment. Answer the questions in the Required Submission section using SQL queries. Remember to note (1) **the answer AND (2) the SQL query** that you used to answer the question as you go in a document for submission.
7. When you are done for the session quite SQLite. I recommend you do not save anything.
8. Close the browser. Repeat steps 1 and 6 until you are done with your homework. You should probably quickly repeat steps 2-5 to refresh your memory each time before going to step 6.

Required Submissions

1. Midpoint
 - a. Confirm you were able to use the software to write a simple SQL query by answering the following questions: What tables do you have in kum.db? How many doctors are at this clinic? What SQL query did you use to find this out?
2. **Assignment 8 SUBMISSION:** Answer the following questions AND submit your SQL queries that were used to find the answer. That is in a separate text file, copy and paste on final SQL that answers the question, then answer the question and move to the next question. **One text file of all your SQL queries and answers to questions are what you submit.** NOTHING else is submitted. You must use SQL queries to answer all questions. If you cannot write your SQL queries, still submit the answers to the questions by reading the tables for partial credit.

At the TOP: List group members, if any, you worked with. But remember, you **MUST** do your own typing.

CAUTION. If you changed the database by accident, stop and exit from everything. And start over so that you are using the correct database. Remember to delete kum.db and upload again. This is a small database, so you should be able to answer the questions manually without writing the SQL queries. You might want to think of this assignment as writing the correct SQL queries to get the same results as your manual answer, noting that you could have made a mistake in your manual answer. So if your SQL query result is not matching your manual answer, think about why and figure out which was wrong. Note that conversely, even if you get the exact same answer as your manual answer the SQL query might not be fully correct. That is, given the particular DB, you got the correct answer, but for a different DB it might not be correct.

Q1: Write a query that displays an alphabetical list (by last name) of all patients.

Q2: Write a query that displays the names of all patients who are Hispanic. Save this as a view to use later.

Q3: Using your saved query (i.e., view) from Q2, display the names of all patients who are Hispanic ordered by last name

Q4: Write queries to answer each of the following questions (you can write as many queries you need to answer the questions):

- 1) How many visits did Lila Autry have? Who saw her on each visit? (*Hint: which columns, from which tables do you need to answer this question?*)
- 2) Who is her primary care doctor (the variable **primary_dr** holds this information) ?
- 3) What were her diagnosis (all of them)? (*Hint: You will have to use the lookup table to find the meaning of the diagnosis code*)
- 4) Were any medications prescribed? If so, how many and what?

Q5: Write a query that displays patients who visited in the second quarter of 2012 together with the number of visits that quarter. Order the output by the number of visits (patients with the largest number of visits should be displayed first). Each patient name should appear once.

Hint: Join tables "patients" and "visits" and use Group By and Count. You'll also need to use a Where for the date range and Sort on (Order by) the Count.

Q6: Write a query that displays the **billed**, **covered**, total paid (**=copay+pat_pd+insur_pd**), and unpaid amount (**=covered-total paid**) for each patient. Save this as a permeant view so that you can use this again in the next task.

Hint: You'll need to calculate new columns for the total paid amount and the unpaid amount. You'll also have to Group By and Sum.

Q7: Using the view from Q6, write a query that displays which patients have not paid their billed amount yet, and what is the outstanding bill amount? Order the output by outstanding bill amount with large amount on top.

Hint: Join tables "patients" with the view and sort. You'll also need to use a Where to find the patients with outstanding bills. Try using greater than 1 cent.

Appendix: Installing SQL on your computer

Using SQLite3 with a Portableapp (to install on any computer including lab machines)

If you do not want to use the cloud tool or it is not working, here is another option. Portableapps are software that do not need to have administrative rights to install and use on a computer. For the most part, you download onto any folder on the computer and install into any folder, then just run the software from there. You can even "carry" your software around on a USB drive (or on your dropbox) and run the software on any computer directly from those locations. If you feel that the software is slow, you might want to make a local copy of the full folder and run it from the computer.

The following only run on windows.

1. Go to: portableapps.com/apps/development/sqlite_database_browser_portable
2. Click on green icon to Download
3. Install into a local folder (e.g. desktop or usb drive). You can copy and paste the folder once installed
4. Run SQLiteDatabaseBrowser (extension is .exe)
5. A text editor portableapp: http://portableapps.com/apps/development/notepadpp_portable

For Mac users, please follow the steps below to download the software:

1. Go to: sqlitebrowser.org/
2. Please download the 'SQLCipher Mac.dmg' by clicking the fifth blue icon on the right side.
3. Find the downloaded software in the 'Downloads' and move to a folder you prefer to use.
4. Double clicks or click 'Open' to run the 'sqlitebrowser_sqlcipher-3.8.0v5.dmg'. There may be a pop-up window asking you whether you sure you want to open it. Please click 'Open', and then you can play with SQL.