Final Project Part 4

Medical Practice Database

**Team: Alex Bordanca, David Thiriot**

for DSCI-D 532 Applied Database Technologies, Spring 2023

*Section 1*

**Application URL**

Our Medical Practice Database web app is hosted at shinyapps.io. It is available at two different links, one under each of our names. Both accounts use the limited, free plan, and providing two links is a hedge against hitting a time and access limit on the free account plan.

Link #1:  [**https://alexbordanca23.shinyapps.io/DB\_Shiny\_Project/**](https://alexbordanca23.shinyapps.io/DB_Shiny_Project/?_ga=2.60446381.1475661514.1682542249-1391138073.1682542249)

Link #2: URL <https://dsthiriot.shinyapps.io/MedPracDB_final/>

Full GitHub URL:

We have uploaded the application source code and database file to github at the following URL.

**https://github.com/AlexBordanca23/DB\_Shiny\_Project**

*Section 2*

The **purpose** for our app is to demonstrate a minimal interactive database suitable for keeping track of information on patients and doctors at a small medical practice. It was also an opportunity to use skills learned during the semester in the course, including designing and implementing a SQL type database with an interactive user interface. We **built** our database using a combination of R and Python. The database is based on SQLite and can be accessed and modified using either R or Python. We wrote the code for the app interface in R using the Shiny App approach.

The **data** in the database is data that we created through simulation in R. We simulated 1000 patients with realistic personal information (names, addresses, emails), together with health information and financial information. This novel dataset became the basis for 7 normalized SQLite tables in our database.

Our web app has the following CRUD **functionality**, accessed through different tab panels:

* View patient information drawn for multiple SQLite tables.
* Update patient health information related to 11 different health conditions.
* Update patient height, weight, heart rate, and blood pressure.
* Visualize (plot) a patient’s height, weight, heart rate and blood pressure in the context of all patients at the medical practice, aggregated separately by biological sex.
* Check and update financial balance, by adding new charges or crediting payments made.
* Update assigned doctor
* Update address and email information
* “Delete” a patient by flagging them as not a current patient. (However, at a medical practice, their information must be retained for a period of time.)
* Add a new patient at the medical practice.

An advantage of using SQLite is that it is self-contained and doesn’t require an additional server connection to a SQL database. The limitations of SQLite did not pose a problem for us based on the size and functionality needed for our database. When the App is deployed in Shiny, we needed to have the SQLite database together in the same folder as the Shiny App, and publishing from R Studio to shinyapps.io was a relatively straightforward and convenient process.

*Section 3*

Contributions and Teamwork

**Alex Bordanca**

There were a handful of different data sets that we had explored in the conception stage of this project, but found that many of them didn’t fit the bill, either in being too large or too small, or by not having real world use cases in terms CRUD functionality. For those reasons, we decided to look at a simulated dataset that David had created. My initial thought on this was to write the app in Django, which I had some familiarity with. Branching ahead, we had a almost deployment ready version of the app relatively early on, although there were a few significant bugs/deficiencies — namely, how quickly the size of the project and number of html files that needed to be written, rendering of foreign key referenced objects for CRUD functionality, and interactive visualization that would have required more knowledge of JavaScript than either of us had or had time to learn. Therefore, I was pleased when David suggested using R shiny to write the app. This was a superior choice, and I feel we both worked extremely efficiently and went from a shell to deployment ready inside of 10 days. We both largely contributed to the success of this project, with major contributions being my experience with git, where I was able to store and branch my own changes, and implement the ones David had sent me via email, and the initial development of the visualization dashboard. Overall, David and I work well together, as we have in the past, and I would gladly look forward to working with him in the future should the opportunity present itself.

**David Thiriot**

After reviewing several possible data sets, I decided to attempt to simulate a novel dataset that would work better for our purpose. I wrote R code to use functions such as ‘charlatan’ and ‘randomNames’ and incorporated realistic health information to simulate a dataset of patients, saved as csv files. We worked together to use Python to create a SQLite database with 7 normalized tables from the simulated data, and demonstrate CRUD functionality with our database. After considering multiple different options for a user interface, we decided to go back to R and create a shiny App. We worked on that together, each writing sections of the Shiny App code. At the time of deployment, we unexpectedly found that we had an app that worked perfectly when run locally on a computer but had an error when running on the shinyapps.io server. After trying several things to troubleshoot unsuccessfully, we consulted with a Shiny expert about this problem and learned that some of the R shiny calls need to use javascript syntax rather than R syntax. We made very subtle changes to the code (for example, changing TRUE to true and ‘$’ to ‘.’ in a conditionalPanel call) to get the App running successfully on the shinyapps.io server. In my view, Alex and I each contributed significantly to this project and worked together well. Beyond our common skills in database design learned during this semester, Alex contributed additional familiarity with github and I contributed some initial familiarity with R. We learned things from each other by working together.

*References*

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Damian W. Bertbenner (2021). randomNames: Function for Generating Random Names and a Dataset. URL https://cran.r-project.org/package=randomNames.

Chamberlain S, Voytovich K (2022). \_charlatan: Make Fake Data\_. https://docs.ropensci.org/charlatan/ (website) https://github.com/ropensci/charlatan (devel).

Chang W, Cheng J, Allaire J, Sievert C, Schloerke B, Xie Y, Allen J, McPherson J, Dipert A, Borges B (2002). \_shiny: Web Application Framework for R\_. https://shiny.rstudio.com/

Muller K, Wickham H, James DA, Falcon S (2023). \_RSQLite: SQLite Interface for R\_. https://rwqlite.r-dbi.org, https://github.com/r-dbi/RSQLite.

R Special Interest Group on Databases (R-SIG-DB), Wickham H, Muller K (2022). \_DBI: R Database Interface\_. https://dbi.r-dbi.org, https://github.com/r-dbi/DBI

Shiny for Python. https://shiny.rstudio.com/py/index.html