

# Beyond SQL

Adrianna Holden-Gouveia

Website: <https://aholdengouveia.name>

**in:** aholdengouveia

 aholdengouveia

## Objectives:

1. Demonstrate how to organize and structure data into tables based on a schema.

## Complete the following problems

References, a video, a PowerPoint and some notes are available at my website <https://www.aholdengouveia.name/AdvData/BeyondSQL.html>

This week's lab can be done using any tool you like that can create a database. You'll be taking the data set from previous week's labs, and turning it into the start of a database. You should upload your data to a cloud database, or use an application on your own computer, or create a virtual machine that has a database installed, such as SQL. Whichever option you choose is fine, just make a note of it, and how you did the setup. Any option you pick must have the ability to share the database, whether that's through an export, or online website or something else. You must have the ability to save your database and share it.

You should take the schema you designed in week 4, and add your data to your database including the tables, keys, and relationships that you designed. This is likely to have several tables, each table needs a primary key.

One way you could do this lab is using <https://www.sandboxsql.com/> we've used before, it's technically SQLite, which is a full implementation of SQL, it's exportable, and because you're working with small enough data sets it should be fine. The only caution I have if you go that route, is make sure you are saving/exporting your data after you make your changes, I do

not control how they do backups, and you shouldn't assume your data will persist after working with it. Make sure you use good version control for this, such as GitHub or GitLab or other version control.

Keep your naming scheme reasonable, you can use the traditional numbering such as Database1.0 Database2.0 Database3.0 for large changes, and Database1.1 Database1.2 Database1.3 for smaller changes. You may use other naming options, but don't do the nonsense of databaseFinal Database-finalFINAL DatabasefinalfinalReallyFinal type of thing.

## **Thinking about your data set, answer the following questions**

1. What might your data be used for in terms of making money?
2. What other data might be collected to enhance your data set? Give at least 5 concrete examples.
3. Make a pro/con list or table for your data set done in a relational database vs non-relational. Make sure you have at least 3 pros and 3 cons for relational AND non-relational
4. If you had to turn your data set into a non-relational model, which one would you pick and why?
5. If you had to use a non-relational model, what other data would you need collected and what ideas do you have for collecting it? Have at least 5 ideas for data and data collection.

## **Dataset into a database**

1. Which tool did you use to make a database? Or did you create a virtual machine with a database tool on it? Why did you pick what you did?
2. What command(s) did you use to make your tables? Did you use a tool/reference to help? If so, which one and why?
3. Screenshot of your open database using the software of your choice that isn't a spreadsheet

4. How long did it take you to get your data into a database? What do you think would make it faster next time?
5. How much cleanup of your data set did you have to do before moving it to the database?
6. Did you have to cleanup the data once it was in the database tables?

## **Deliverables**

- An image of your schema
- Your actual database
- Answers to the above questions clearly numbered so it's obvious which question you're answering and which section