Why SQL + R is an affable combo when I start learning SQL?

Joshua Marie

2025-11-28

Table of contents

1 Introduction

I am already using R since 2018, and uses SQL since 2022-2023. Way back in 2023, I am learning one of the most valuable feature in R, and that's the ability to integrate R into other software. That's because I only use softwares independently, i.e. R only, Python only, etc. This is how I first learn SQL, and I learn few frameworks that integrates R and SQL databases.

If you've spent any time in data science, I am sure you encountered language wars and such debates — there's like hundreds or maybe thousands of blogs spread in the community comparing which languages is better or worse. I am not talking about that in this blog post, however, here's the thing — it's not really a versus situation. SQL and R are like peanut butter and jelly. Each is good on its own, but why not both? *Flavorful*.

Regardless, SQL excels at what databases do best: storing, organizing, and retrieving massive amounts of data with lightning speed. R, on the other hand, shines where creativity and

complexity matter: statistical modeling, advanced visualizations, and transforming raw data into insights that actually mean something.

In this post, I'll show you why combo-ing R and SQL isn't just nice to have — it's my stack. And more importantly, I'll show you what I know how to make them work together so seamlessly you'll wonder how you ever worked any other way.

2 Tools and Packages

I can name few tools and packages on working with SQL databases in R, most especially when you just started. I don't have any database in my own device, but did you know you can simulate databases? These are the tools and packages to start:

1. `{tidyverse}` — Why this? This is a package that holds the complete set of tools in data science, and that includes working with databases. Speaking of which, this is a *meta-package* that also contains what we need: `{dbplyr}`, which also contains `{DBI}` package dependency.
2. `{box}` — I already talked about this package in my previous blog posts. Please, take a look at them if you have some time:
    - Box: Placing module system into R
    - In my "Ways to load / attach packages in R" blog post
3. `{dbplyr}`
4. `{DBI}`
5. `{RSQLite}`

3 With existing database

I learn SQL thanks to SQLite. This is a language-agnostic library, written in C, that can act like a database while being lightweight, and you can use it literally everywhere! It is also used to built into everywhere, it could be mobile phones and most computers.

Thanks to SQLite, I made a first move to learn SQL without installing heavy softwares, such as PostgreSQL and MySQL, just to learn SQL. Additionally, SQLite is an open-source, but not open for contribution (I believe this is designed for good purpose).

In a positive sense, R and SQL is a great combo. Maybe R and SQL is not a great combo for software development as Python and SQL combo, R and SQL can make a place in data analysis instead. As long as you have `{DBI}` and `{RSQLite}` installed in your R, you can now make a first move on integrating R and SQL, and you're good to go.

3.1 SQLite in R

The only primary requirements are `{DBI}` and `{RSQLite}`. If you know how to write a query, you don't need a compatible set of packages in `{tidyverse}` and `{dbplyr}`, otherwise, as long as you know how to use `{tidyverse}` packages, namely `{dplyr}`, `{tidyr}`, etc., you can use it instead.

4 Learning SQL and R combo without a server

But I know some of you wants to know what it looks like to use the existing database and then call it in R. I literally said in the introduction that you can *simulate* — I have another different meaning: use `simulate_*` family functions in `{dbplyr}`

5 Conclusion

SQL and R aren't competitors—they're collaborators. SQL is your data retrieval expert, getting you exactly the data you need with incredible efficiency. R is your analysis specialist, turning that data into insights, models, and visualizations.

The data scientists who succeed are the ones who can speak both languages fluently. They use SQL to ask databases the right questions, and R to find the answers that matter.

So don't pick sides. Master both. Your future self (and your future employers) will thank you.