

Let's Build a Simple Database

Writing a sqlite clone from scratch in C

[Overview](#)

[View on GitHub \(pull requests welcome\)](#)

How Does a Database Work?

- What format is data saved in? (in memory and on disk)
- When does it move from memory to disk?
- Why can there only be one primary key per table?
- How does rolling back a transaction work?
- How are indexes formatted?
- When and how does a full table scan happen?
- What format is a prepared statement saved in?

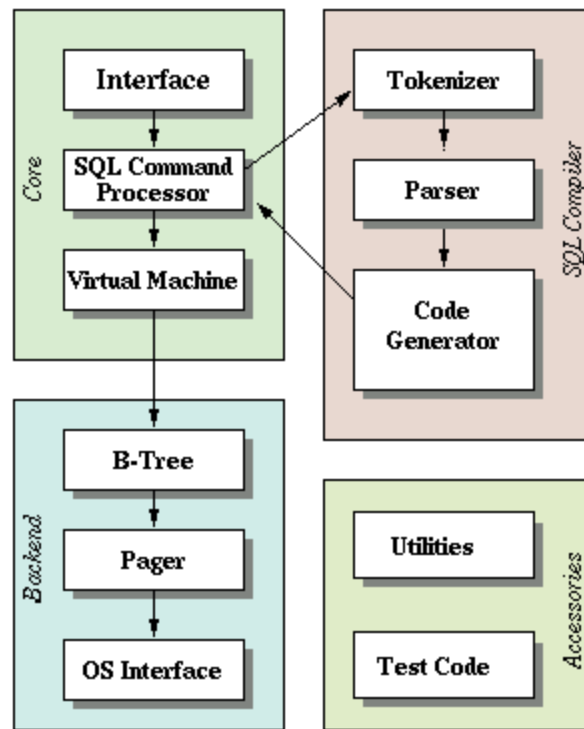
In short, how does a database **work**?

I'm building a clone of [sqlite](#) from scratch in C in order to understand, and I'm going to document my process as I go.

Table of Contents

- [Part 1 - Introduction and Setting up the REPL](#)
- [Part 2 - World's Simplest SQL Compiler and Virtual Machine](#)
- [Part 3 - An In-Memory, Append-Only, Single-Table Database](#)
- [Part 4 - Our First Tests \(and Bugs\)](#)
- [Part 5 - Persistence to Disk](#)
- [Part 6 - The Cursor Abstraction](#)
- [Part 7 - Introduction to the B-Tree](#)
- [Part 8 - B-Tree Leaf Node Format](#)
- [Part 9 - Binary Search and Duplicate Keys](#)
- [Part 10 - Splitting a Leaf Node](#)
- [Part 11 - Recursively Searching the B-Tree](#)
- [Part 12 - Scanning a Multi-Level B-Tree](#)
- [Part 13 - Updating Parent Node After a Split](#)

"What I cannot create, I do not understand." – Richard Feynman



sqlite architecture (<https://www.sqlite.org/arch.html>)

[rss](#) | [subscribe by email](#)

This project is maintained by [cstack](#)

Hosted on GitHub Pages — Theme by [orderedlist](#)