Wyatt Revalee

R.B. Abhyankar

CS457

4/3/23

M9A1 – SQLite

While there are many different options of databases in today’s world, there are a few more prominent ones. Perhaps one of the most well-known database management systems is Structured Query Language, more commonly known as SQL. There also exist a plethora of libraries that make use of SQL in order to create new database engines. Some examples are PostgreSQL, MySQL, and SQLite, the last being the focus of this report.

SQLite is a C-language library that makes use of the SQL database engine. It is “small, fast, self-contained, high-reliability, full-featured” database engine, and is the most popular of all its peers. Not only is it used by those who are managing databases, it is also built into all mobile phones as well as most computers and comes bundled inside a multitude of apps that are used everyday around the globe. The source code for SQLite is also in the public domain, meaning that it is free to use by anyone for any purpose.

SQLite was designed by Dwayne Richard Hipp in spring of the year 2000 while he was working for General Dynamics on a contract with the United States Navy. It was designed for use in a damage-control system aboard guided-missile destroyers, though now of course, it is used for many other things. The Lite in the name sort of introduces what the engine was designed to be. While most databases require a database management system in order for the user to interact with them, SQLite does not. Instead, the engine is integrated into applications using a linker, allowing it to be accessed using simple function calls inside of the application. No server or administrator needed. SQLite requires no separate server process and reads writes directly to ordinary disk files. It is very compact, with a full-feature-enabled version having a library size of seven hundred fifty kilobytes (750KiB). The more memory given to SQLite, the faster it runs, but even in low memory environments, it performs well. All these features make up the compactness that gives the engine its name.

On top of being compact and easy to access, SQLite also has a reputation for being very reliable. With each new version created, an automated test suite exists that runs millions of test cases that cover the entirety of the engine, in order to weed out any potential bugs that have been created. The engine also responds to any failures or errors well, and transactions remain ACID compliant even if interrupted by a system crash or power failure. The automated suite tests for these as well using simulated cases. Despite all of this, there still exits bugs in the engine. However, documentation of these bugs are open and honest, with bug lists available as well as up-to-date code changes for fixing them. There also exists an international support team behind SQLite that work the engine full-time. The team is constantly working on new updates for the engine, while also being available for professional support for those who request it.

In order for one to get started with SQLite, you would need to download and install SQLite tools onto your computer. First one would need to make their way to SQL’s website and to their downloads page. Then they would select the version they wish to download, which depends on the system they are using. Once installed SQLite can be ran and used from the command line. With the download also comes a GUI for interacting with some parts of the engine easier. There also exists third party tools that allows the user to get more out of their experience with the engine and increase their productivity. These are completely optional but exist for anyone who wishes to use them.

Although SQLite is much easier to install and set up compared to other, more complex database management systems, it still requires proper learning of it in order to be used. There exists tools to help new users to get started with SQLite, such as sample databases. These are premade databases for users to use as a sort of “playground” in order to learn commands and get used to using them. SQLite also contains its own commands for use inside of it known as sqlite3 commands, whose functionality can be compared to shell commands. These commands do all sorts of things, such as display helpful commands to the user, display all current databases, save results of queries, and execute statements from a file.

Like every other database management system, one of the first things to learn how to do in SQLite is make queries. Queries allow the user to interact with data inside of a database and exists as a pseudo-conversation between them and the database – the user asks for data and the database responds with it. The keyword for beginning a query in SQLite is SELECT. The user can then specify from what tables and rows they want their data and filter it however they wish. Sorting and filtering data are also extremely important things to learn in order to use a database effectively and are the next two steps for beginners to learn on their road to mastering SQLite. When a user needs to query data from multiple tables, those tables can be joined using commands. This allows for completely new queries to be made on data that would have otherwise been separate. Furthermore, grouping allows data to be combined with each other in order to better summarize it for reporting purposes. Taking things a step even further, set operators allows results from multiple queries to be combined together. This can be done by combining all results of the queries, only distinct rows from them, or rows that both queries contain into a set.

SQLite also has a wide range of uses that make it a wise choice for those choosing between several database management systems. First, the engine can be used in many different kinds of devices across the board, whether it be cellphones, PDAs, or an MP3 player. With how efficient SQLite is in its use of memory and space, as well as its reliability and no need for maintenance., it also creates a small code footprint. SQLite can also be used in applications for reading and writing to several types of files, like CSV or JSON. This makes things easier for the user as the engine handles a lot of the tedious work for them. For those needing to store data for the backend of a website, SQLite is a great choice. It requires no configuration and stores information in regular files, making it perfect for smaller and medium sized websites.

Although SQLite has many great things going for it, it still has its limitations. The main limitation of the engine is that it was not designed to be used in large-scale applications that require a high level of concurrency. This can cause performance issues in these environments, making other DBMS’s a better choice for larger projects. Due to the compactness of SQLite, it also lacks some features that other DBMS’s do not, such as stored procedures and user defined functions. When it comes down to it, what makes SQLite great for smaller to medium applications, is also what limits it from being as useful for larger applications.

To summarize, SQLite allows developers to implement databases that are both lightweight and fast. As opposed to other databases, SQLite is easy to install and implement, without needing to worry about memory usage or lack in performance. The database is also self-contained and extremely reliable, not requiring any maintenance or even a server with its use. For developers working on small to medium-sized applications, it can be hard to find a reason to not use SQLite. Although it has its limitations, they only exist due to the features that make SQLite great at what it does. With how feature rich and lightweight SQLite is, it is easy to see why it has become one of the most popular database in the world.