

2023 ASIS&T Webinar Series

February 9, 2023

“Getting started with SQLite”

Sponsored by DCMI

webinars@asist.org

asis&t

Association for Information Science and Technology

Getting started with SQLite

A preface for DCMi Education Committee tutorial series on SQLite

Database Systems

Database Systems

Overview of Database Systems



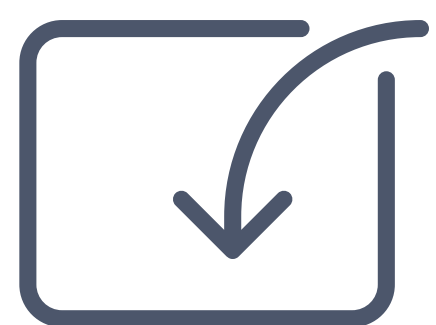
Data Management

Database System is a software that manages data stored in a computer.



Data Access

It provides methods for data storage, retrieval, and modification.



Data Integrity

It ensures data consistency and integrity through transactions and constraints.

Types of Database Systems

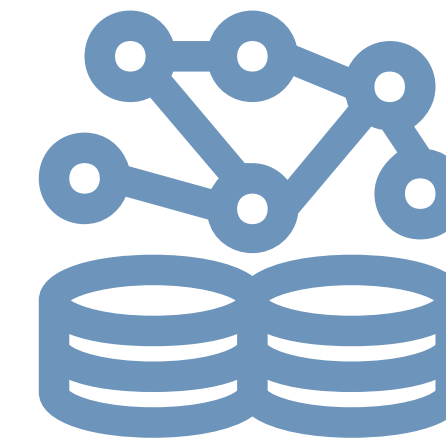
SQL & NoSQL Databases

SQL databases



- Relational databases
- Use Structured Query Language (SQL) to manage data
- Examples: MySQL, PostgreSQL, MariaDB, SQLite

NoSQL Databases



- Non-relational databases
- Use non-SQL data structures to manage data
- Examples: MongoDB, Cassandra, CouchDB

Relational Databases

Popular SQL Databases



Relational Databases

Popular SQL Databases

Server / Client Databases

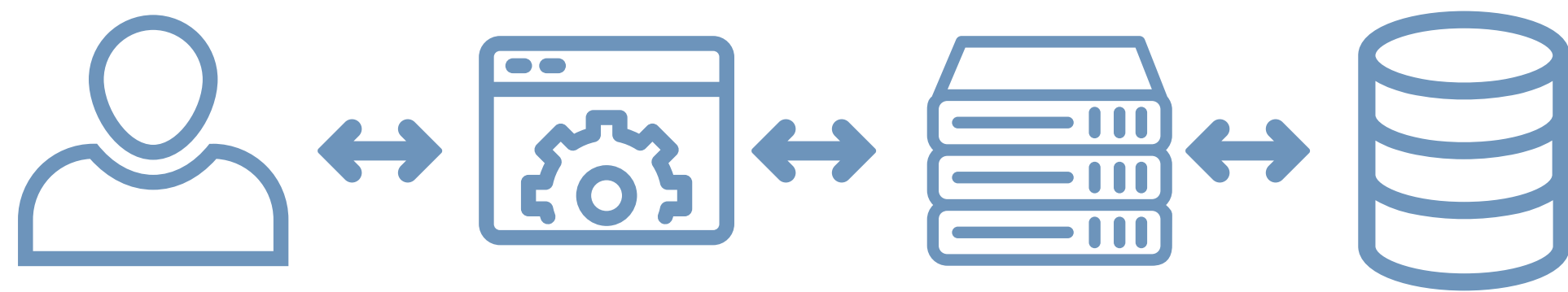


Serverless Database



Serverless Database

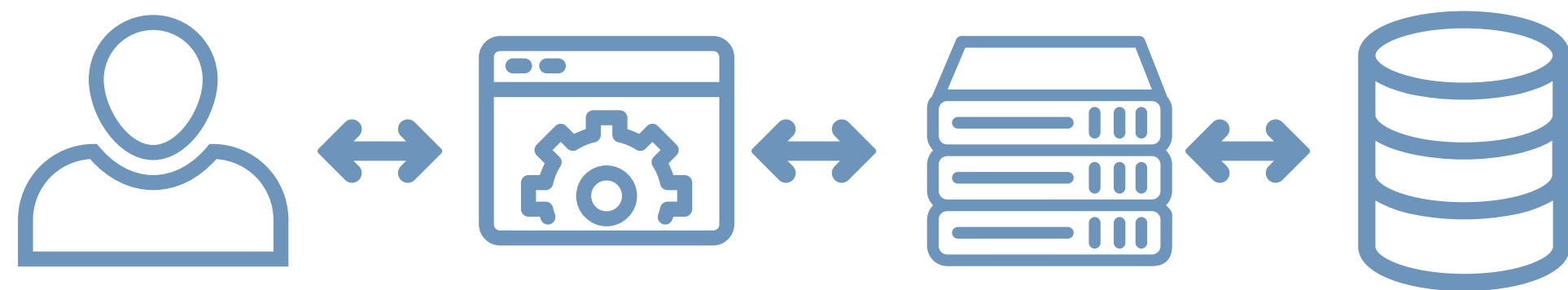
Server / Client Databases



- Users interacts with a client application.
- Clients connects to a database server.
- Database server connects to a database

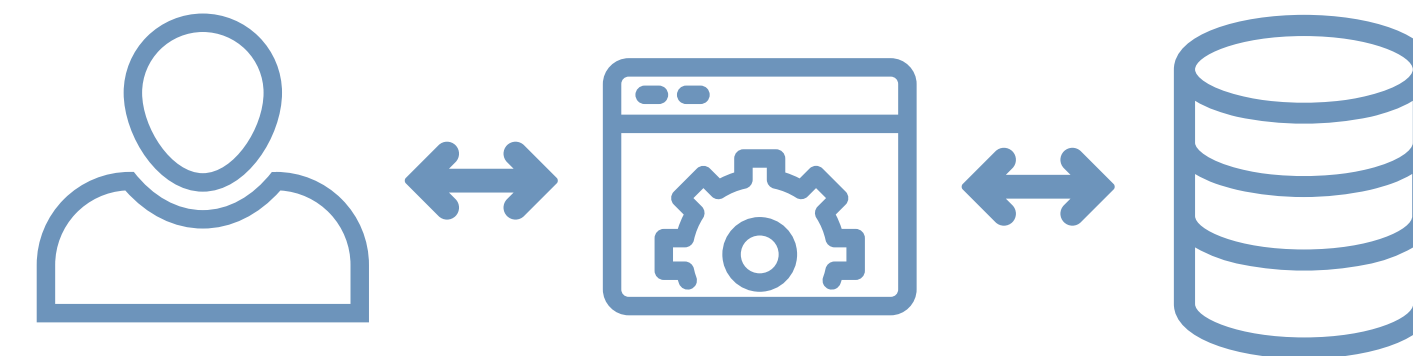
Serverless Database

Server / Client Databases



- Users interacts with a client application.
- Clients connects to a database server.
- Database server connects to a database

Serverless Databases



- Users interacts with an application.
- Application connects to a database / database file directly.

Advantages of server-less databases



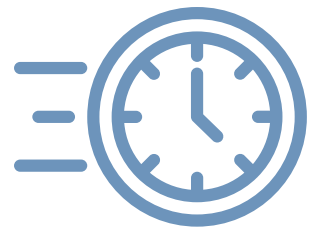
Flexible and portable

Can be deployed and used without the need for dedicated infrastructure, making them flexible and portable.



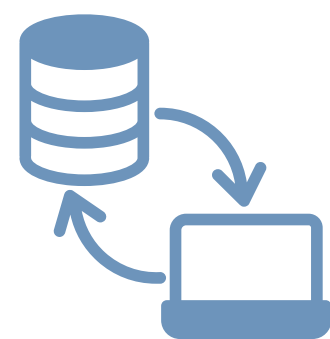
Easy to Maintain

Freeing up the user from infrastructure maintenance, security updates, and other administrative tasks.



Development speed

Can help to speed up development time by eliminating the need to provision and manage infrastructure



Easy Backup & Replication

Usually server less databases are file-based database, that makes them easy to backup and replicate.

SQLite & SQLite file format

History of SQLite

Beginning, design goals and the growth



SQLite was created by **D. Richard Hipp** in **2000**.

SQLite was designed to be a **small, fast, reliable, and self-contained** SQL database engine

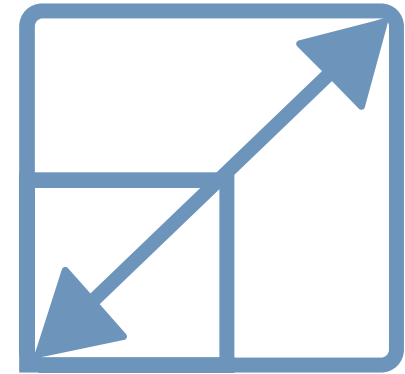
Over the years it has evolved into one of the **most widely-used database systems** in the world.

Features of SQLite

Significant features of SQLite as a database system



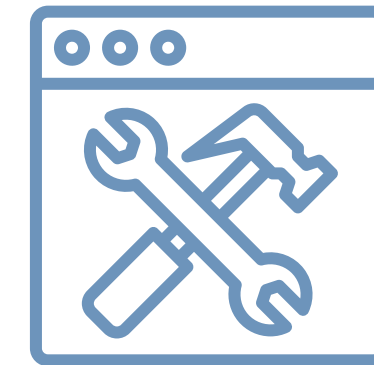
Multi Platform



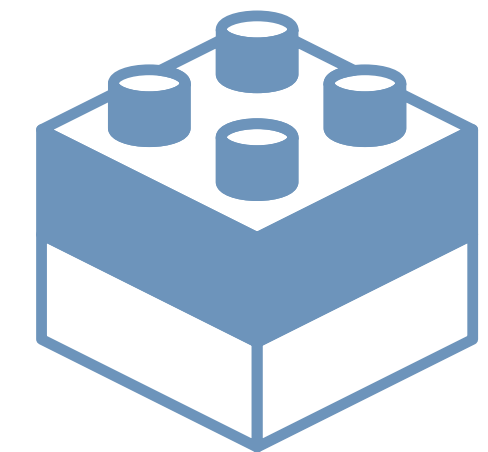
Scalable



Open Source



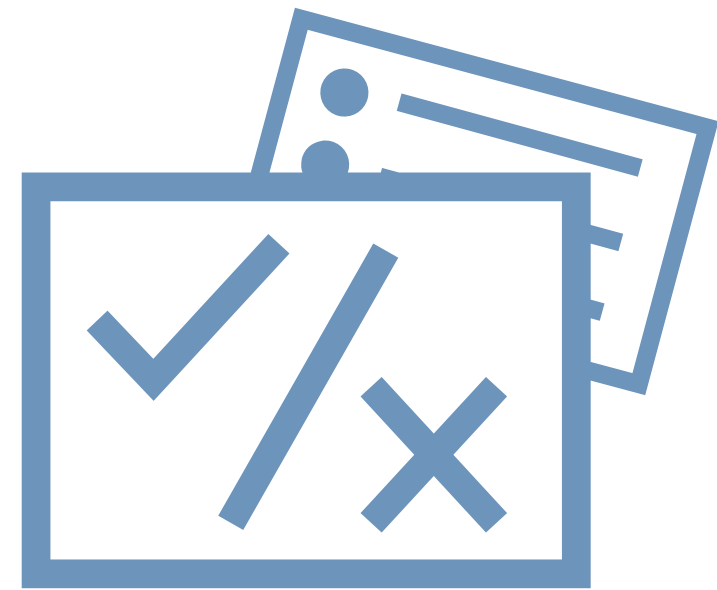
Zero Configuration



Embedable

SQLite: Extensively Tested

One of the most thoroughly tested software projects in the world



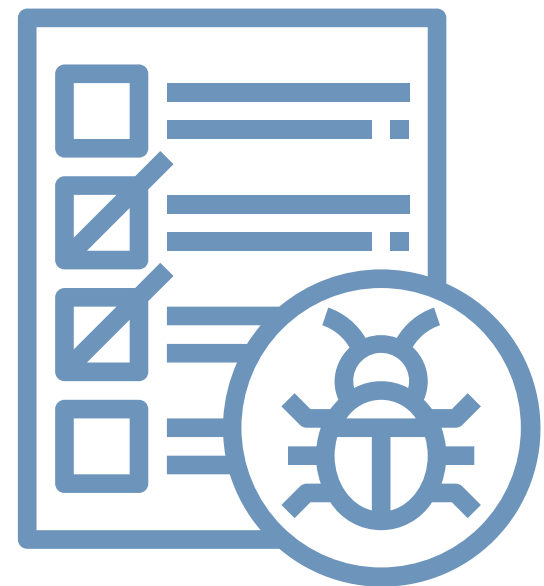
Over **10 million** unit & query **tests**. **2.5 billion tests** in "soak test" before release.

100% statement & branch coverage, including edge cases.

SQLite: Extensively Tested

One of the most thoroughly tested software projects in the world

Low bug count, rare data-loss or corruption bugs. Most bugs are performance-related.



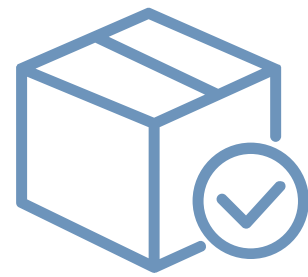
With its extensive testing, SQLite has a **robust and reliable codebase**, making it a dependable choice for a wide range of applications.

SQLite file format



Single file database

Binary file format used to store SQLite databases.



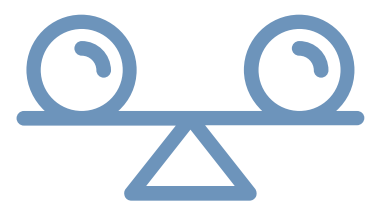
Self-Contained

Self-contained and contains all data and metadata for the database.



Cross Platform

Cross-platform compatibility, can be used on multiple operating systems and devices.

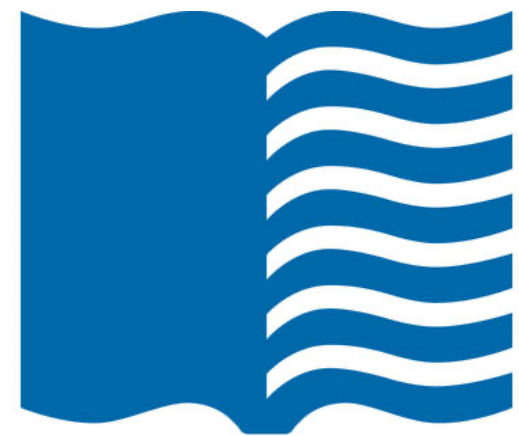


Highly Stable

Highly stable, ensuring long-term support and compatibility.

SQLite and the GLAM community

Use of SQLite and SQLite file format in digital preservation



LIBRARY OF
CONGRESS

SQLite is **recognized by the Library of Congress** as a key technology for **preserving digital information**

SQLite is a **Recommended Storage Format for datasets** according to the US Library of Congress.

Widely used in archives, libraries, and other cultural heritage institutions for data preservation and management

Long term support

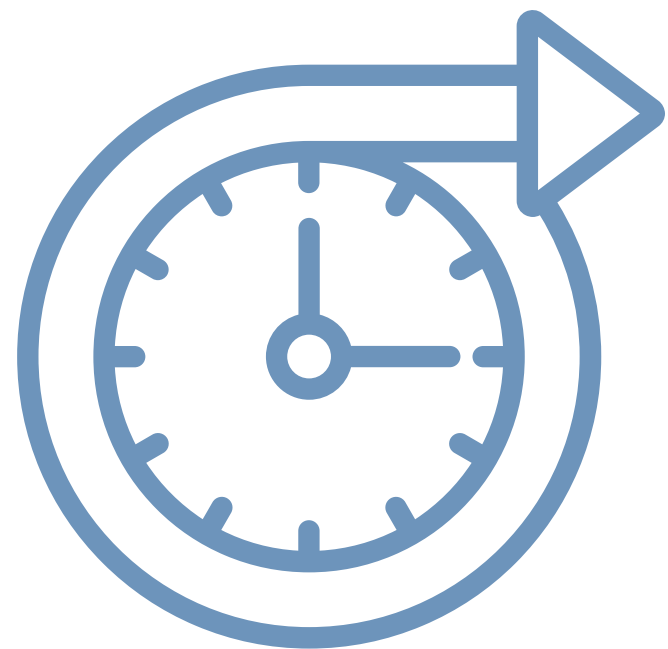
Support for SQLite Library and SQLite format through 2050

Developers aim to **support SQLite through 2050.**

Promise to keep C-language API & on-disk format **fully backwards compatible.**

Applications written today will be able to use **future versions of SQLite** released decades from now.

Ensures continued use and **compatibility** of SQLite **for the long term.**

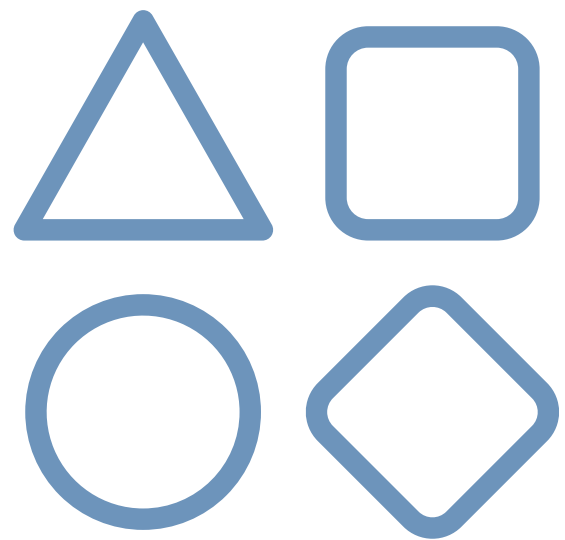


SQLite and Data Types

SQLite rows are typeless, and mixing datatypes is permitted

Unlike other databases, **SQLite does not use strict data types**

Columns are assigned an "**affinity**" of NULL, INTEGER, REAL, TEXT, or BLOB



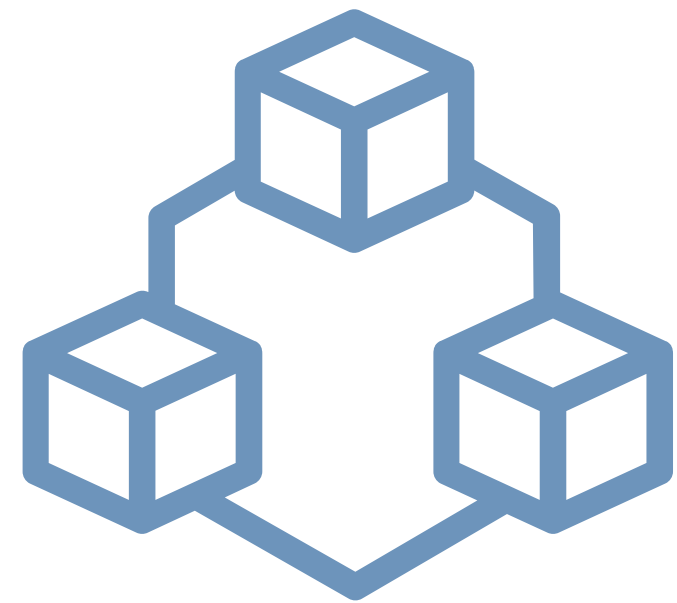
Affinities **help determine usage** in queries and functions

Mixing data types in columns is allowed without issue

SQLite support in languages

Most of the popular programming languages support SQLite natively

**Most programming languages have built-in/
native support for SQLite, including:**



**C, C++, Python, Ruby, Go, Rust, Java,
JavaScript, Lua, Objective-C, Perl, PHP ...**

Basic SQL

SQL

Structured Query Language

Standard language used for managing and manipulating relational databases

Allows to perform operations such as:

- **Creating, altering and deleting** tables
- **Inserting, updating, and deleting** data
- **Retrieving and aggregating** data



Basic SQL Commands

Some of the basic and essential SQL commands



CREATE: creates a new database table

ALTER: modifies an existing database table structure

DROP: removes a database table or its elements (such as columns)



SELECT: retrieves data from a database table

INSERT: adds new data to a database table

UPDATE: modifies existing data in a database table

DELETE: removes data from a database table

Creating a table

SQL for creating a table in SQLite

```
CREATE TABLE books (  
  book_id,  
  title,  
  author,  
  publication_year  
);
```

book_id	title	author	publication_year

Add new data to the table

SQL for adding data to the table

```
INSERT INTO books (book_id, title, author, publication_year) VALUES  
  (1, 'Harry Potter and the Philosopher''s Stone', 'J.K. Rowling', 1997),  
  (2, 'The Shining', 'Stephen King', 1977),  
  (3, 'Murder on the Orient Express', 'Agatha Christie', 1934);
```

book_id	title	author	publication_year
1	Harry Potter and the Philosopher's Stone	J.K. Rowling	1997
2	The Shining	Stephen King	1977
3	Murder on the Orient Express	Agatha Christie	1934

Retrieve data from the table

SQL for selecting data from the table

```
SELECT title, publication_year FROM books
```

title	publication_year
Harry Potter and the Philosopher's Stone	1997
The Shining	1977
Murder on the Orient Express	1934

Conditionally retrieve data

SQL for conditionally retrieving data

```
SELECT * FROM books  
WHERE publication_year > 1990;
```

book_id	title	author	publication_year
1	Harry Potter and the Philosopher's Stone	J.K. Rowling	1997

Basic query composition

Basic structure of an SQL query

```
SELECT * FROM books  
WHERE publication_year > 1990;
```

books

book_id	title	author	publication_year
1	Harry Potter and the Philosopher's Stone	J.K. Rowling	1997
2	The Shining	Stephen King	1977
3	Murder on the Orient Express	Agatha Christie	1934

SQLite Tools

SQLite CLI

SQLite provides a powerful CLI tool

Easy to use and lightweight.

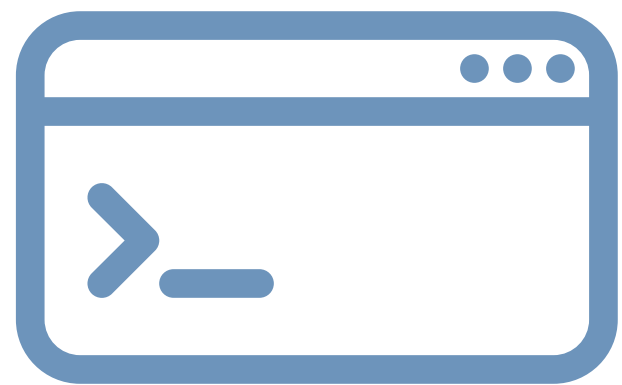
Cross-platform compatibility.

Supports SQL syntax for database creation, manipulation, and query.

Supports reading and writing data to and from disk.



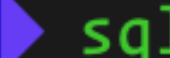

Can be used to manage multiple databases.

Accept parameters and Dot[.] commands.



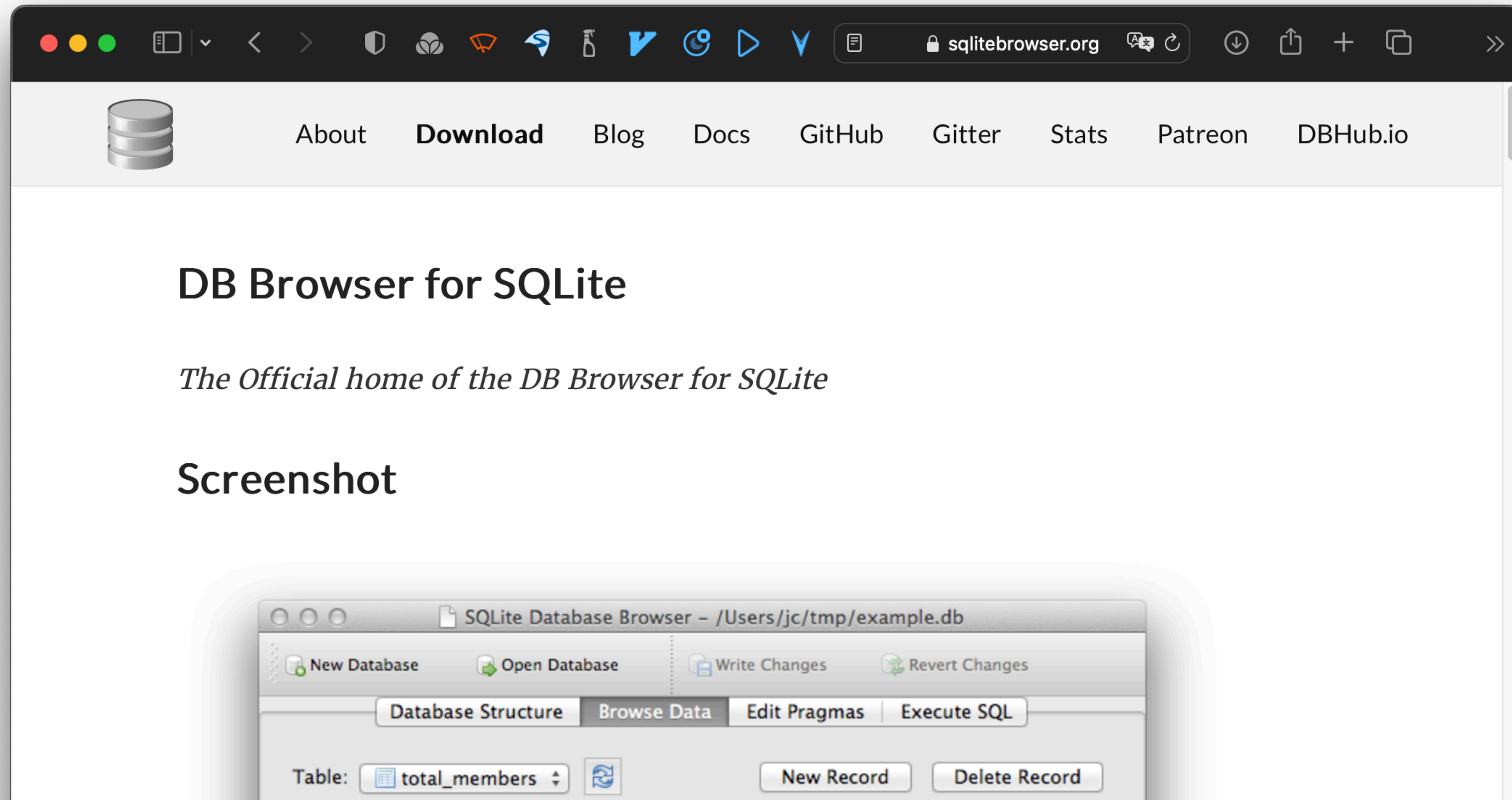
SQLite CLI

SQLite provides a powerful CLI tool

```
(base)   ~/Desktop  sqlite3 books.db
SQLite version 3.39.3 2022-09-05 11:02:23
Enter ".help" for usage hints.
sqlite> .table
books
sqlite> .schema
CREATE TABLE books (
  book_id,
  title,
  author,
  publication_year
);
sqlite> SELECT * FROM books;
1|Harry Potter and the Philosopher's Stone|J.K. Rowling|1997
2|The Shining|Stephen King|1977
3|Murder on the Orient Express|Agatha Christie|1934
1|Harry Potter and the Philosopher's Stone|J.K. Rowling|1997
2|The Shining|Stephen King|1977
3|Murder on the Orient Express|Agatha Christie|1934
sqlite> 
```

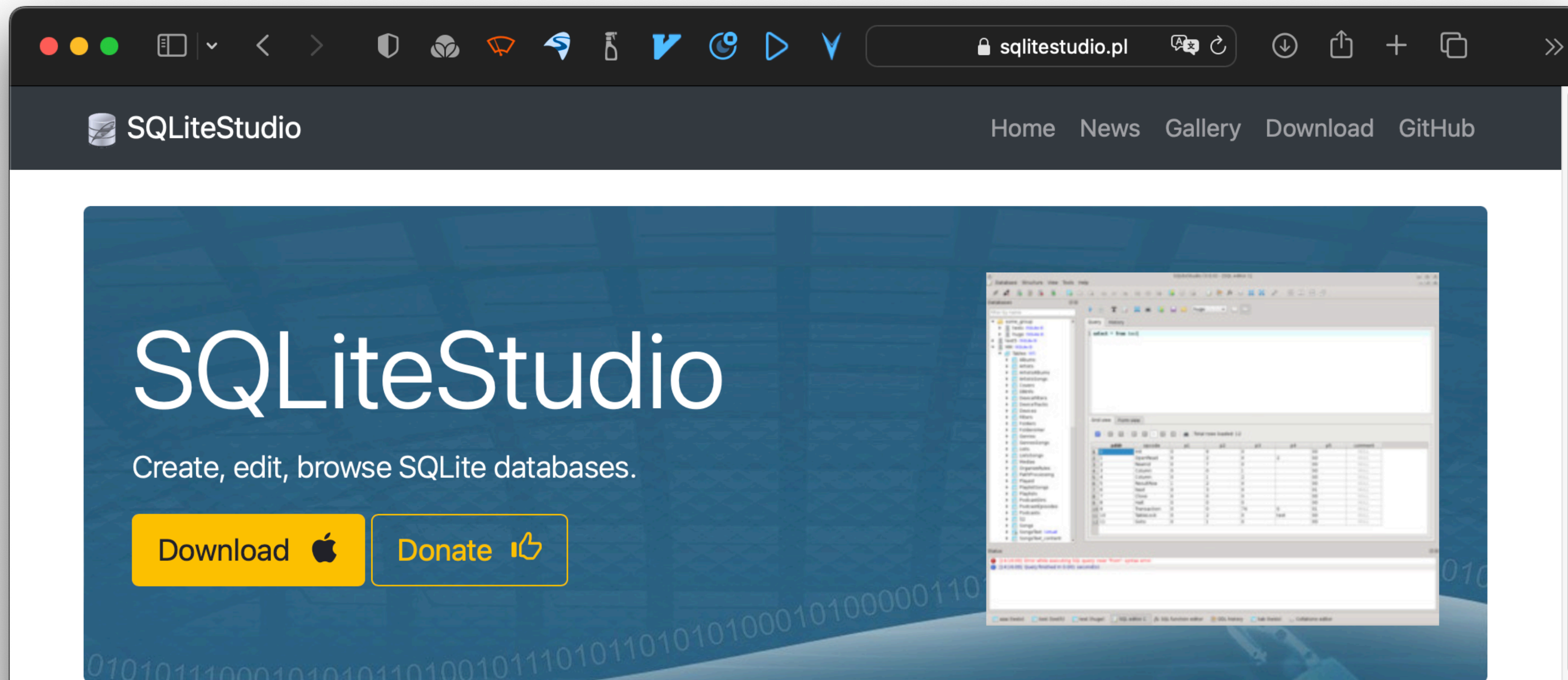
DB Browser of SQLite

www.sqlitebrowser.org



SQLiteStudio

www.sqlitestudio.pl



3.4.3 released!

DBeaver

www.dbeaver.io

DBeaver Community
Free Universal Database Tool

Star 30,345 | Follow @dbeaver_news

search here ... Go

Home About Download Sources Documentation News Support DBeaver PRO CloudBeaver

Universal Database Tool

Free multi-platform database tool for developers, database administrators, analysts and all people who need to work with databases. Supports all popular databases: MySQL, PostgreSQL, SQLite, Oracle, DB2, SQL Server, Sybase, MS Access, Teradata, Firebird, Apache Hive, Phoenix, Presto, etc.

The screenshot shows the DBeaver application interface. On the left, a tree view displays the database structure. The main window shows a table with columns 'column1' through 'column8' and a grid of data. A 'Mock Data Generator' dialog box is open, allowing users to configure data generation settings for the selected columns, including data types, sequences, and templates.

SQLite for Web

SQL.JS

sql.js.org



The screenshot shows a browser window with the URL `sql.js.org/#/`. The page title is "SQL.js" and the main heading is "SQLite compiled to JavaScript". The left sidebar contains a navigation menu with items: "API documentation", "Usage", "Demo", "Examples", "Inside the browser", "Example HTML file:", "Creating a database from a file chose...", "Loading a database from a server using fetch", and "using XMLHttpRequest". The main content area features a yellow "SQL .JS" logo, a title "SQLite compiled to JavaScript", and status badges for "CI passing", "npm v1.8.0", and "cdnjs v1.8.0". The text describes `sql.js` as a JavaScript SQL database that runs in the browser, uses a virtual database file stored in memory, and allows importing and exporting SQLite files. It also mentions the use of `emscripten` for compilation and the inclusion of contributed math and string extension functions.

SQL.js

SQLite compiled to JavaScript

- API documentation
- Usage
- Demo
- Examples
 - Inside the browser
 - Example **HTML** file:
 - Creating a database from a file chose...
 - Loading a database from a server
 - using fetch
 - using XMLHttpRequest

SQLite as Webassembly

Evolving SQLite based Web dev ecosystem



WebAssembly (Wasm) is a binary instruction format for a stack-based virtual machine, designed as a portable compilation target for programming languages, enabling deployment on the web for client and server applications.

Tutorial Series on SQLite

Follow [@Dublincore](https://twitter.com/Dublincore) or visit www.dublincore.org for updates

THANK YOU FOR ATTENDING



Submit a webinar proposal at
<https://www.asist.org/meetings-events/webinars/>



webinars@asist.org

A copy of the recording and a follow-up survey
will be emailed within 24 hours.

asis&t

Association for Information Science and Technology