Final Project: MySQL Versus SQLite

Grace Stewart and Dae'onna Butler

CS200: SQL and Relational Databases

Dr. Rylan Chong

December 1, 2023

Description

SQLite is a server-less database management system that implements a compact form of a similar database management system: MySql. SQLite is a project developed by D. Richard Hipp which began on May 5, 2000. It is a self-contained, C-language library that is less than 750KiB in size, which is comparably much smaller than other database management systems. According to the SQLite website, it is the “most widely deployed database in the world” and includes several “high-profile projects” (Hipp 2023).

Key Features Compared to MySQL

Sqlite	MySql
Fully featured MySql with advanced capabilities	Fully featured MySql
Developed using C programming language	Developed using both C and C++ programming languages
Open-source project	Licensed
Dynamic Schema	Fixed Schema
Does not require a server to run	Requires a server to run

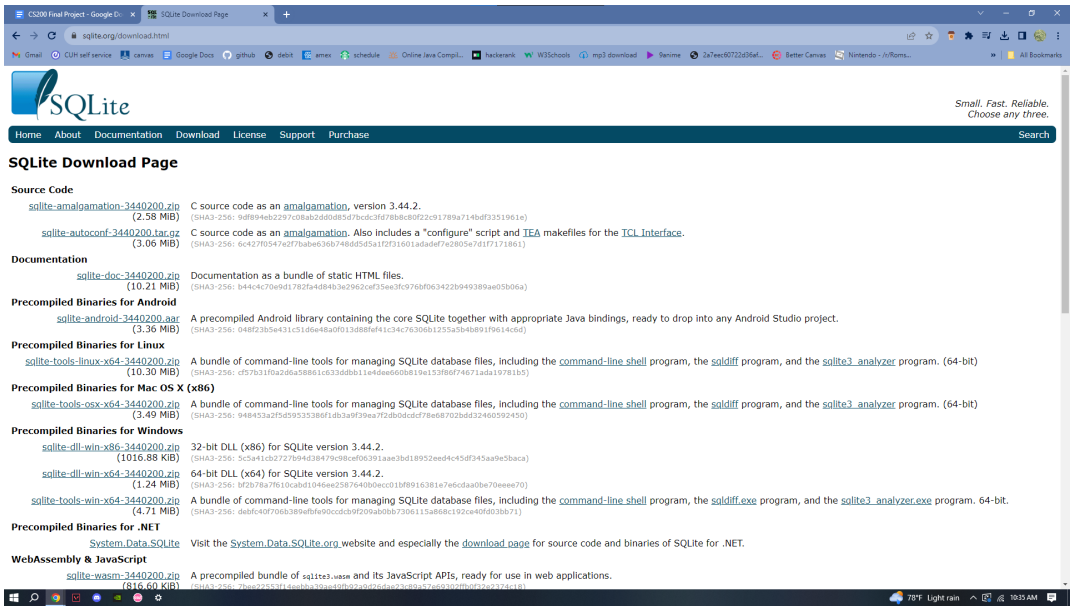
Compared to MySQL, SQLite has many key features that differentiates it from the former. For example, SQLite claims to have “advanced capabilities” compared to MySql. These capabilities include, but are not limited to having “partial indexes, indexes on expressions, JSON, common table expressions, and window functions” (Hipp 2023). Adding on, while MySql utilizes both C and C++ programming languages, SQLite only uses C. In addition, this database engine is an open-source project, allowing anyone to use the code for any commercial or public setting. SQLite also features a dynamic schema compared to MySql that uses a fixed schema.

This refers to the fact that the schema, also known as the structure of your database, is able to change simultaneously as data is inputted. As mentioned previously, SQLite is also serverless. Being serverless has many pros and cons. While having a server can provide “better protection against bugs in the client application,” as well as being able to more accurately control and access the database, being serverless also has its own number of advantages (Hipp 2023). One instance being that there is no separate server process needed to “install, setup, configure, initialize, manage, or troubleshoot the database” (Hipp 2023). This means any program that is able to access the disk is able to use SQLite. Because of features such as its licensing, scalability, and data model, SQLite proves to be a strong competitor amongst many database management systems.

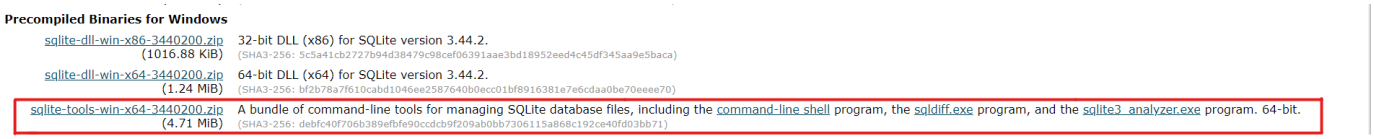
Installation and Setup

PC:

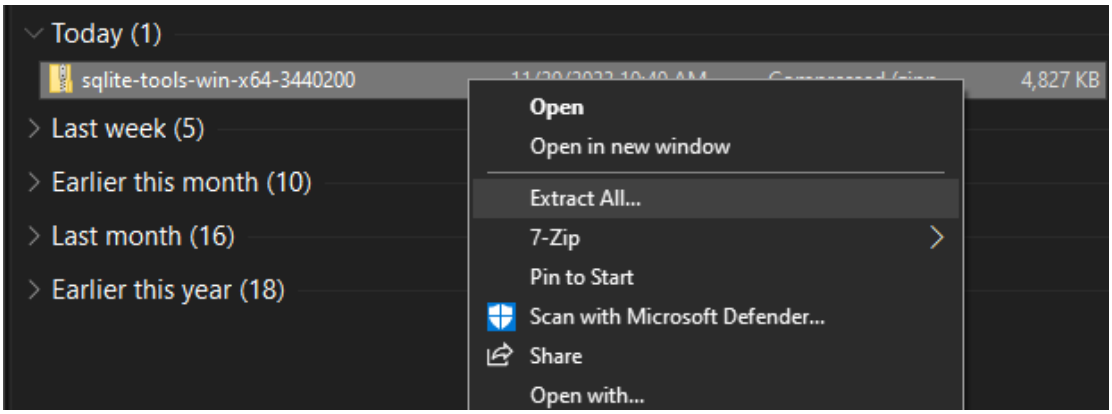
1. Begin by going to the Sqlite download page: https://www.sqlite.org/download.html



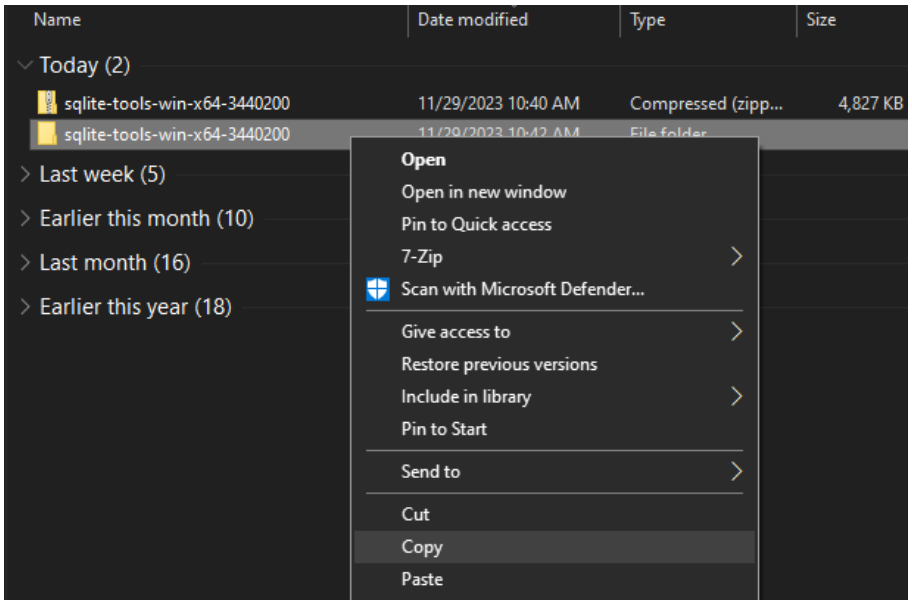
2. Click on the third download under “Precompiled Binaries for *Windows*.” It should be titled “sqlite-tools-win-x64-3440200.zip”



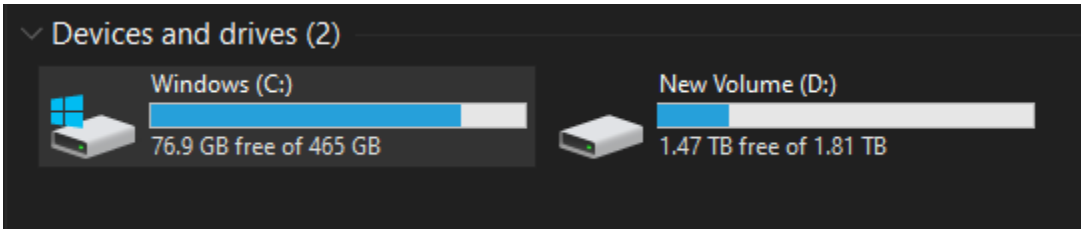
3. Once downloaded, right click on the downloaded file, extract all the files,



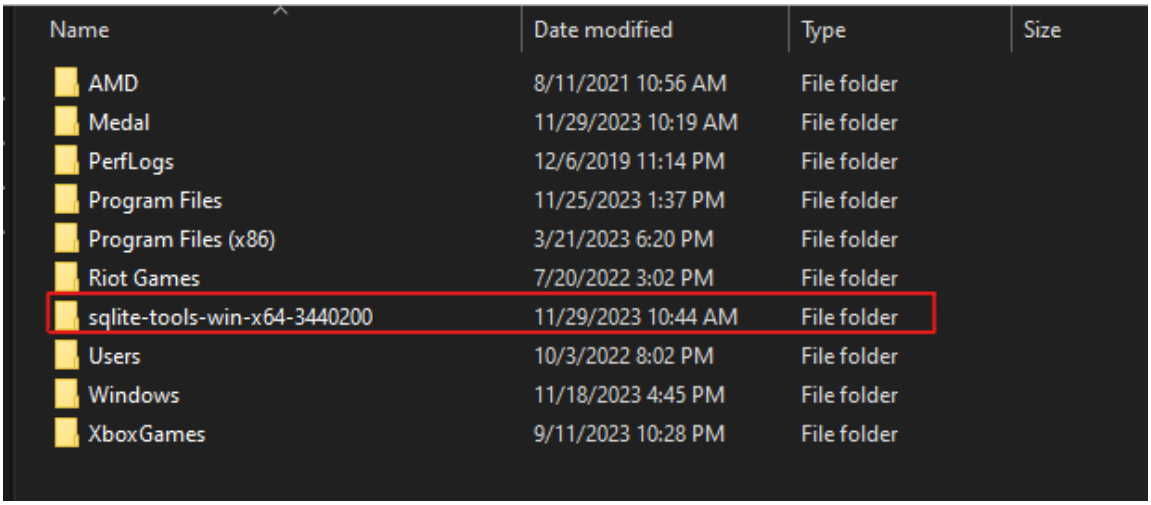
and copy the extracted files.



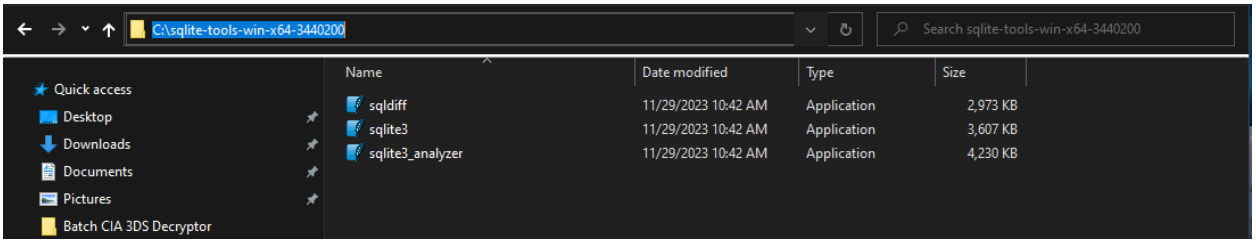
4. Next locate your c:drive.



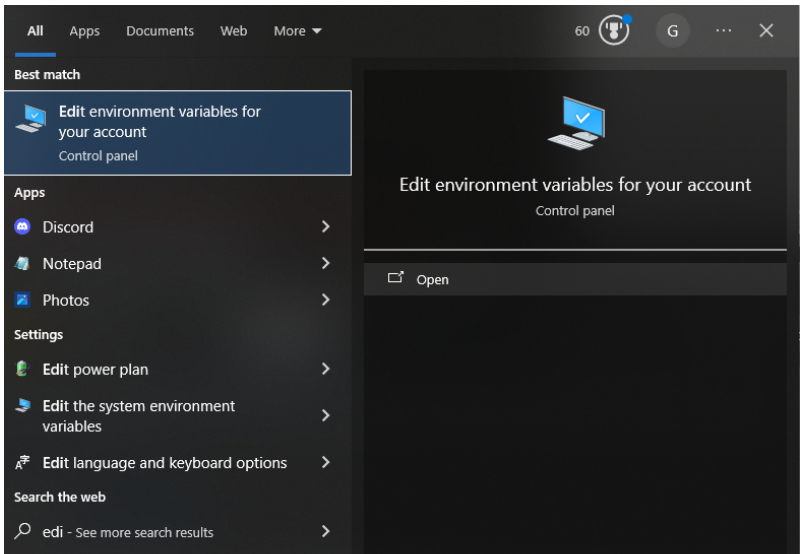
Paste the files into the drive,



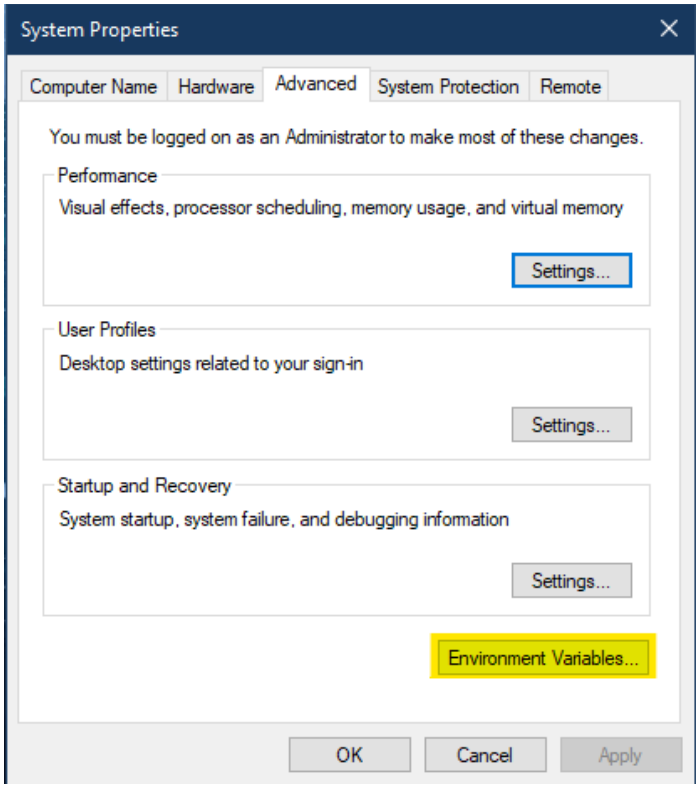
and copy the c:drive file's location.



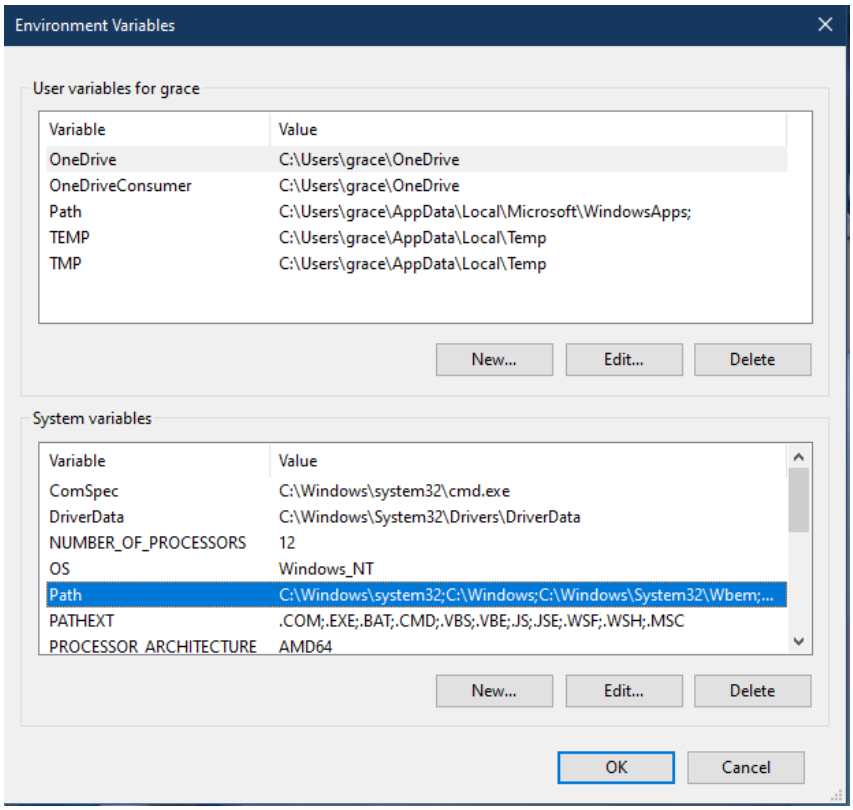
5. Search for the control panel named "edit the system environment variables" in the search bar. Click "open."



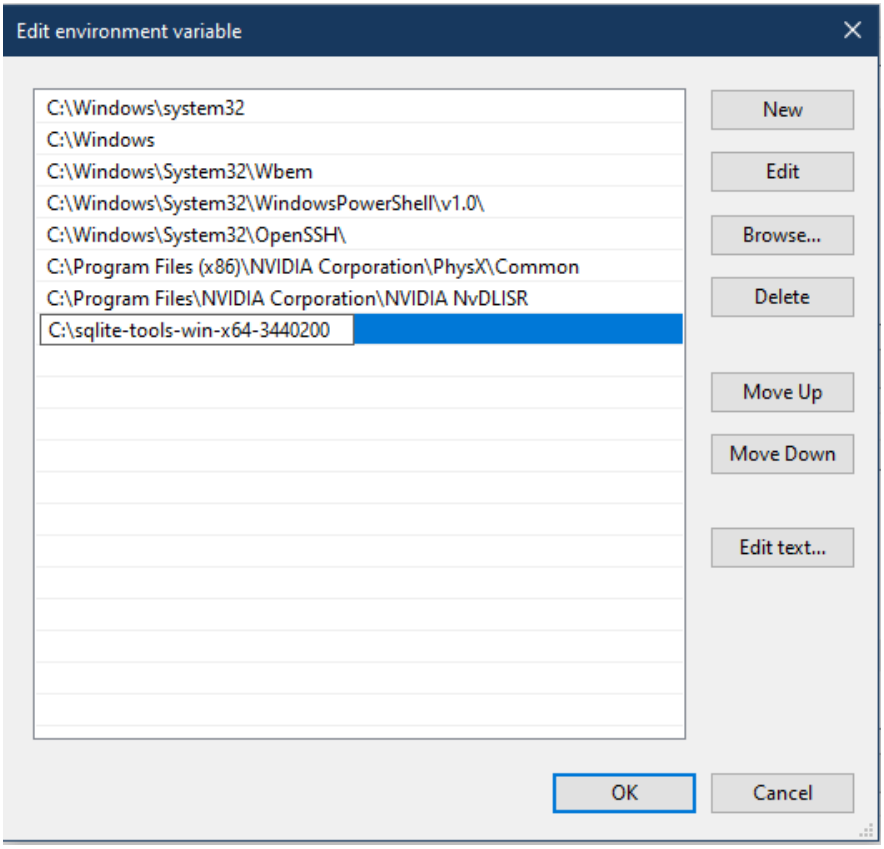
6. Once the system properties tab is opened, click on “Environment Variables.”



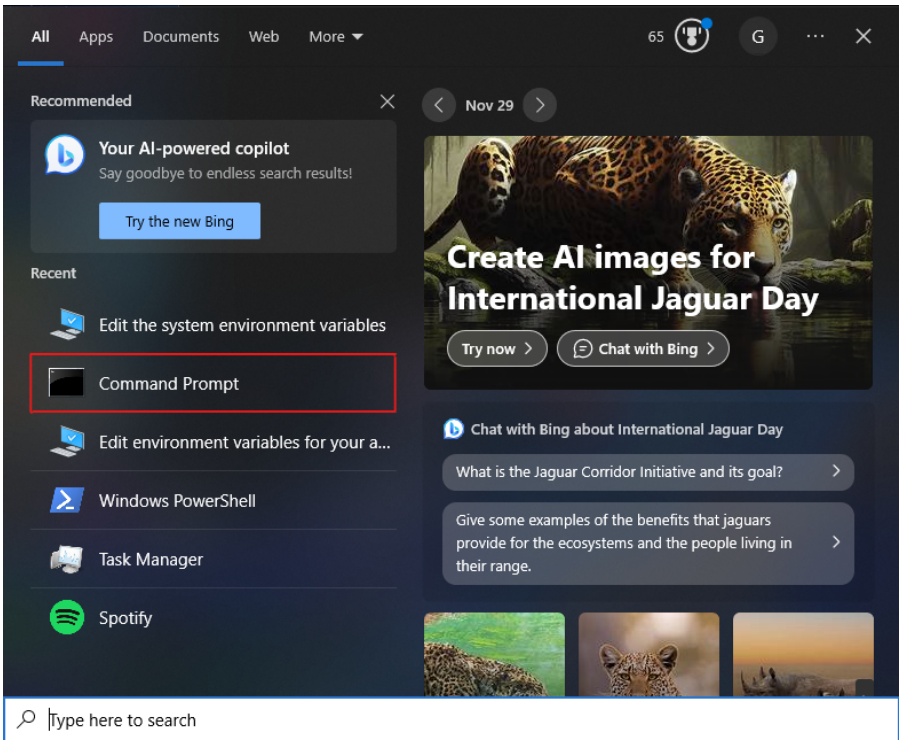
7. Once you are in Environment Variables, double click on “Path” located under “System Variables.”



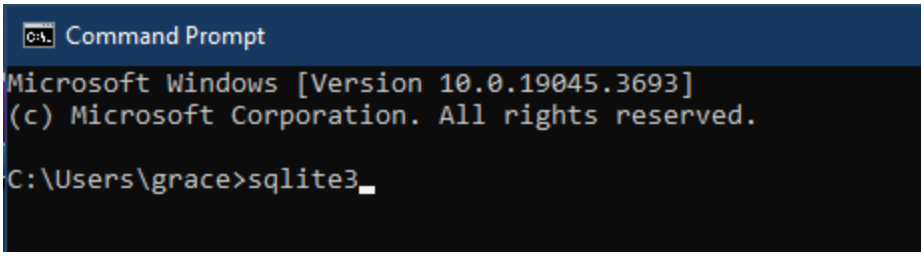
Then, click “new” and paste the copied c:drive file location into the path.



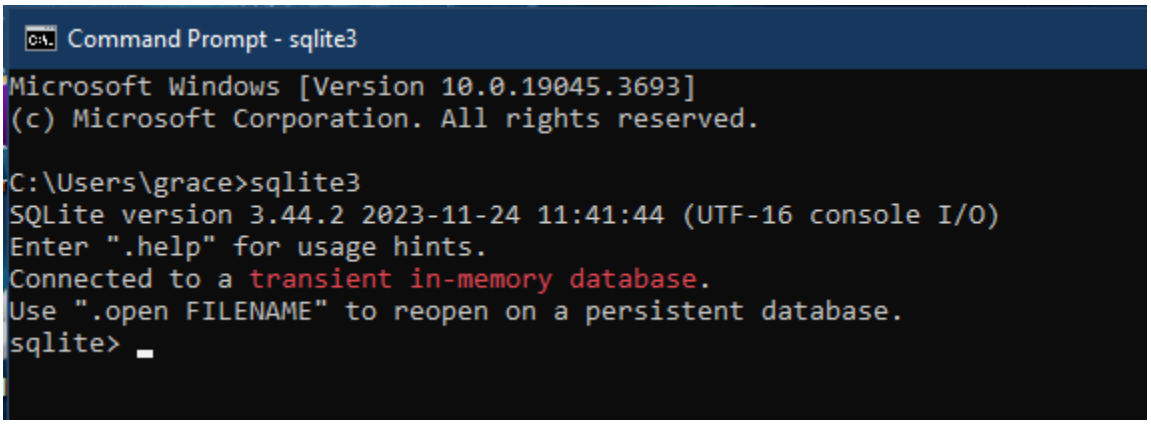
8. Open CommandPrompt.



Type Sqlite3 and hit enter (or return depending on your keyboard layout).

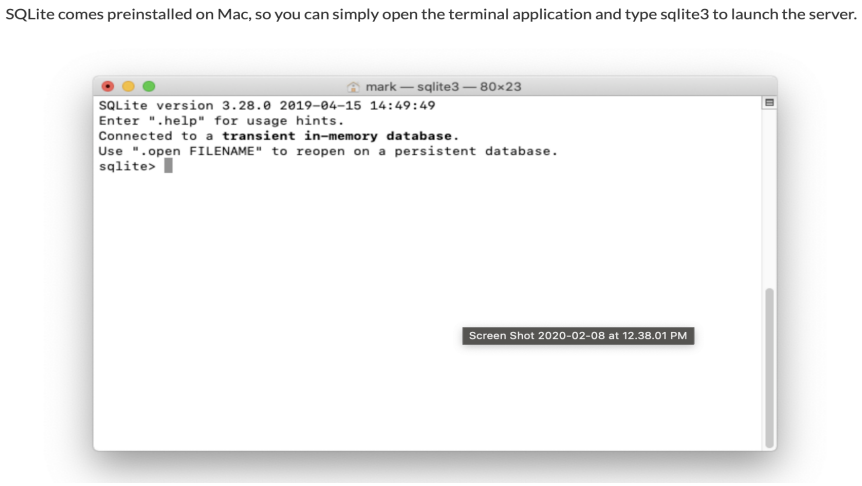


Once you see this text appear, you have completed the Sqlite installation for windows.



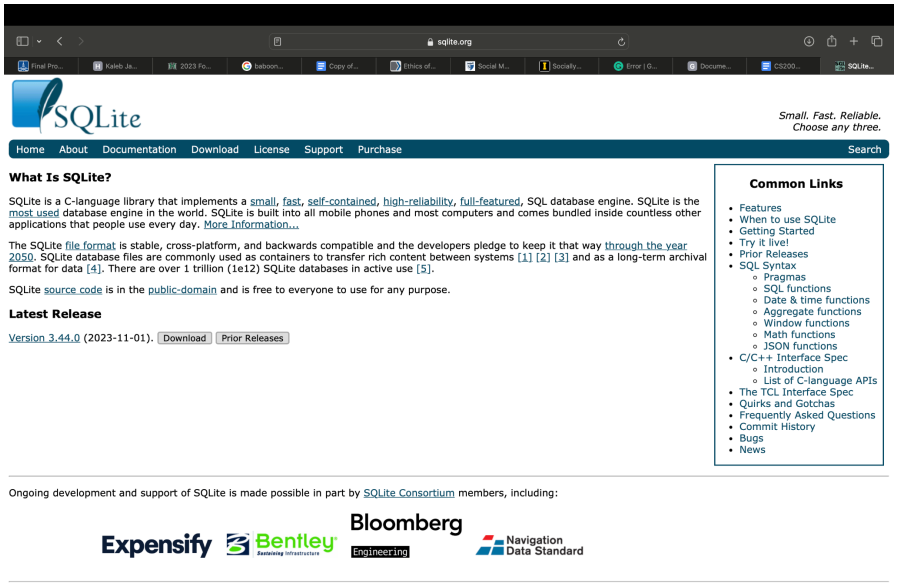
Mac:

Most Mac systems already come with SQLite preinstalled

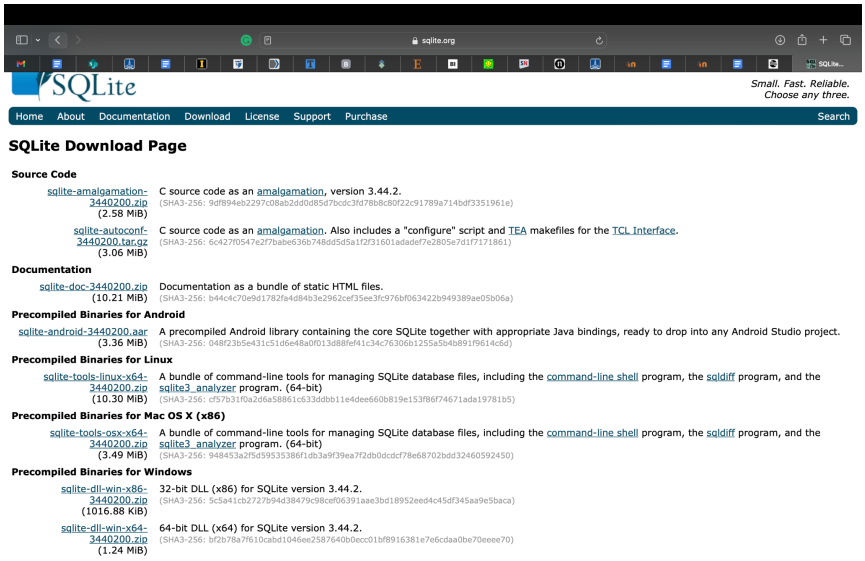


However, in the case that it is not, here are the steps to do so:

1. Begin by going to the Sqlite download page: https://www.sqlite.org/download.html

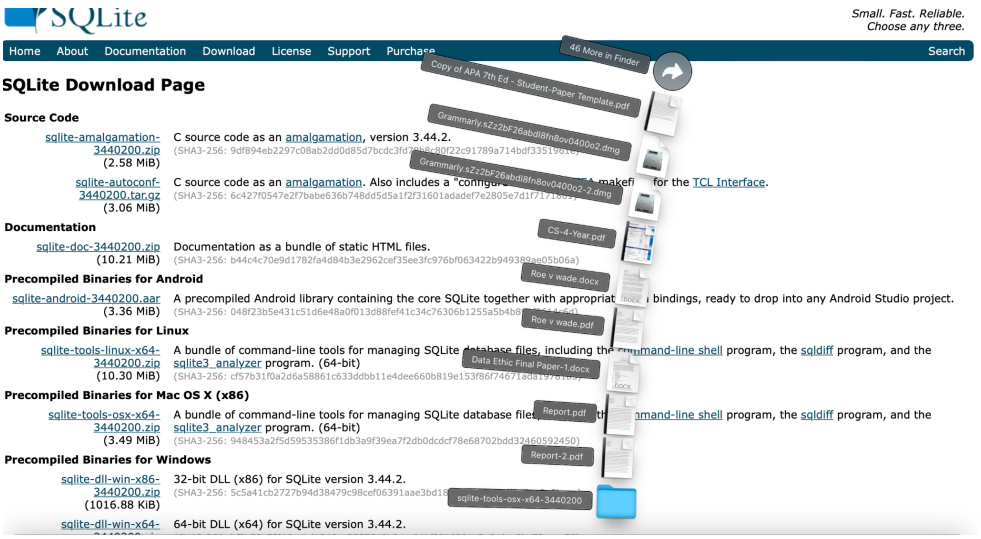


2. You will then be guided to this page, which includes a number of different links to different versions of SQLite. (note: you only need to focus on your OS)

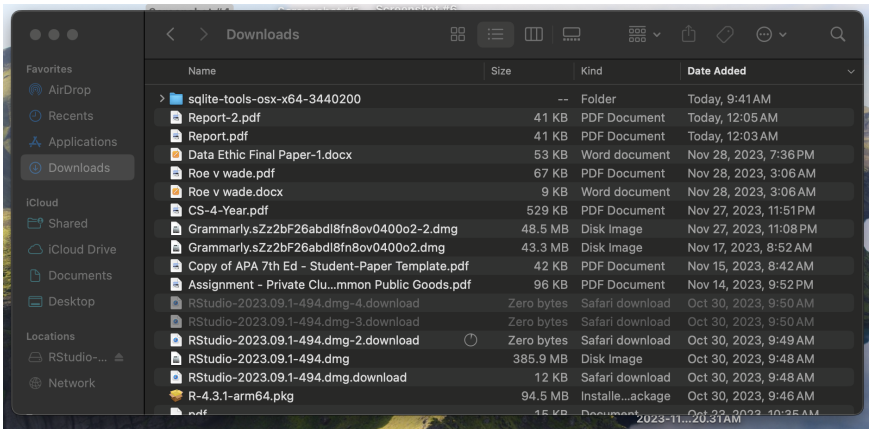


3. Click on the download under precompiled Binaries for Mac OS

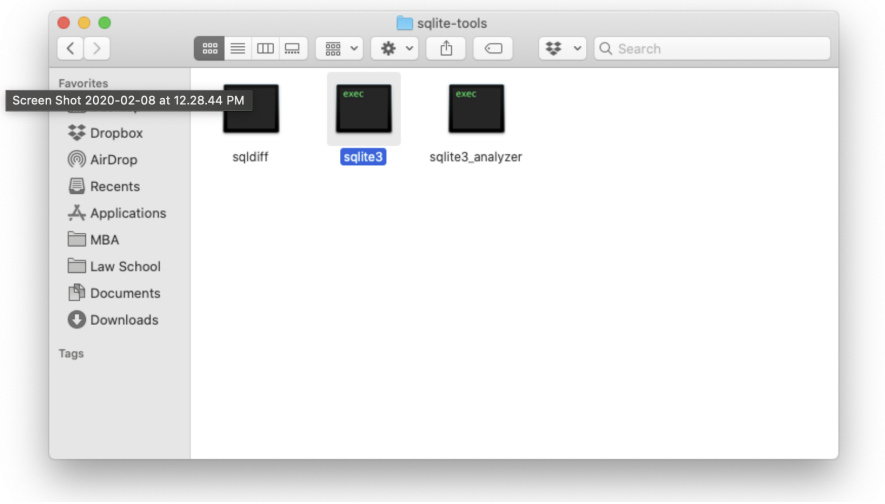
4. It should then be the most recent download located within your downloads folder



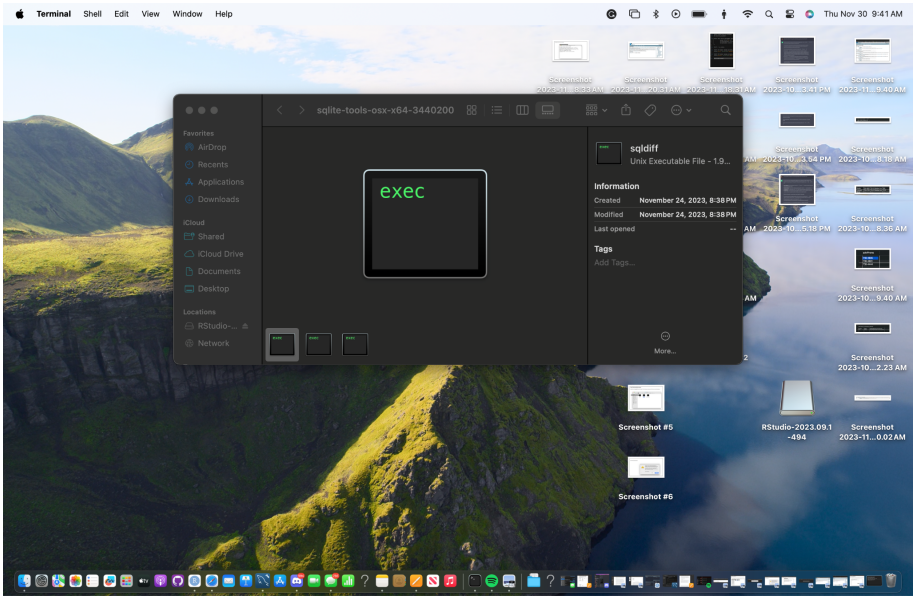
5. You should be able to view the downloaded file in your “downloads” folder.



6. After opening, you will be presented with 3 different SQLite options.



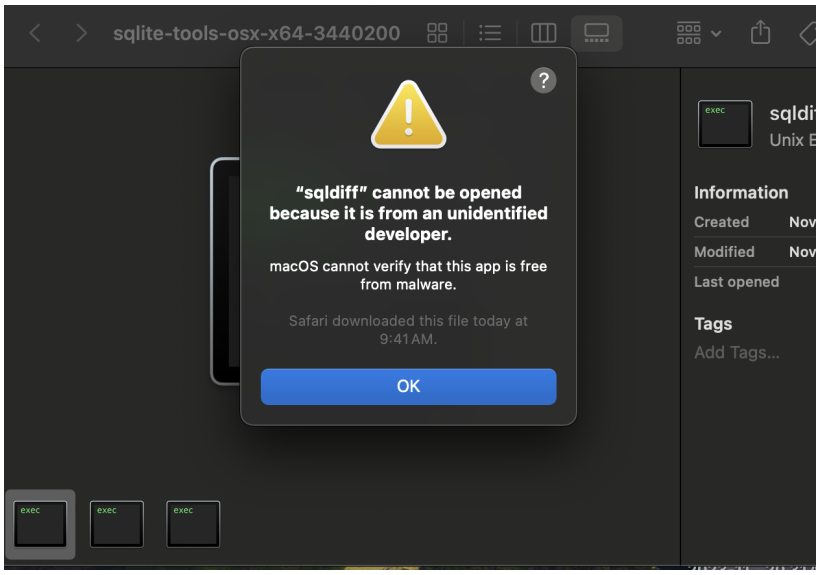
7. Right click on the “SQLite3” file and click open.



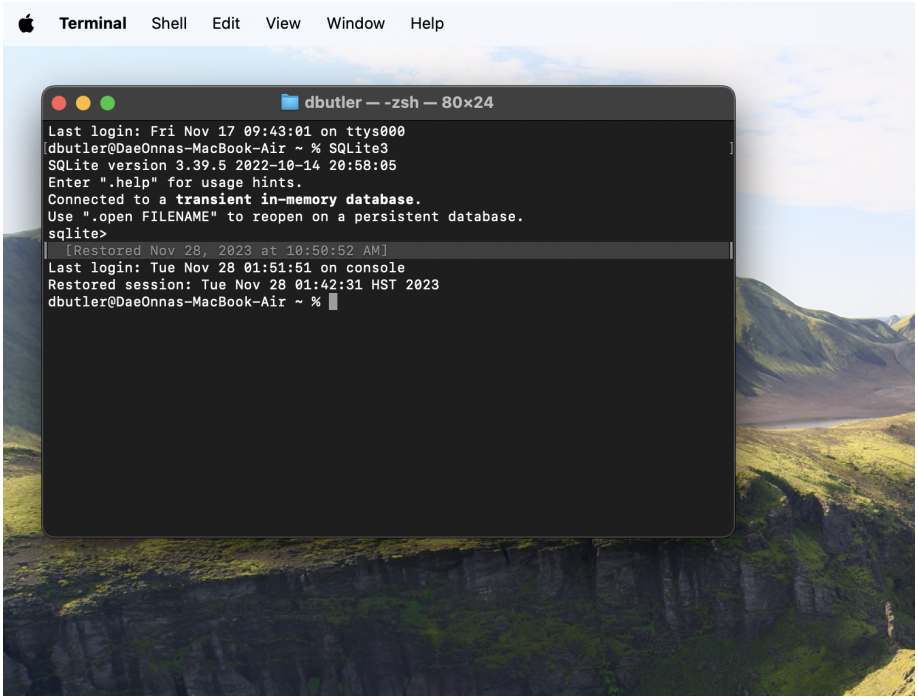
8. When done properly, you will be presented with this screen and you can click “open” (you may skip step 9 if this step appears).

Click the Open button:

9. If this is the screen that appears, then there is a very likely chance that SQLite is already installed on your Mac and in which case you can just use the finder tool to open the terminal.



10. Type Sqlite3 and hit return (or enter depending on your keyboard layout).



11.

The server can be quit in Mac OS X by using the **.quit** command.

Code Cases

1. Creating your database:

Once you are in the terminal, you will begin coding. To create a database, use the following simple syntax:

```
.open practiceDB.db
```

* The "." is used to change the output format of queries or to execute prepackaged query statements.

* ".db" is the file type

CODE CASE #1:

2. Creating a table:

To create a table, you are going to begin by following the template. In each “DATATYPE” slot, you are going to type in the corresponding data type to the column you are creating. This means that if your column’s values will be associated with a number value, you will utilize the INTEGER data type. Primary key values uniquely identify a column in the table.

It is important to note that you must always follow the last line of code/end of your statement with a semicolon (;) in order to signify the end of the code you are going to run.

```
CREATE TABLE table_name
        (column1 DATATYPE PRIMARY KEY,
        column2 DATATYPE,
        column3 DATATYPE,
        column4 DATATYPE,
        column5 DATATYPE);
```

List of data types:

NULL- null or empty value

INTEGER- signed integer

REAL- floating point value

TEXT- text string

BLOB- “blob of data,” kept exactly how it was inserted

Foreign keys reference columns and values from other tables. When utilizing foreign keys, first begin by making sure foreign keys are enabled within your Sqlite database with the following code:

```
PRAGMA foreign_keys;
```

If the command returns the value “0” they are disabled. If it returns “1,” it is enabled. To enable them, type the following:

```
PRAGMA foreign_keys = ON;
```

To disable foreign keys, type:

```
PRAGMA foreign_keys = OFF;
```

To add a foreign key to your table, insert this line of code at the end of your table. You must make sure the foreign key’s data type matches across all tables.

```
FOREIGN KEY (column1) REFERENCES table_name (column1)
```

To drop a table, you can use the command:

```
DROP TABLE table_name;
```

Practice creating the following tables:

```
CREATE TABLE games
        (title TEXT NOT NULL PRIMARY KEY,
```

```
    studio TEXT NOT NULL,
    releaseYear INTEGER NOT NULL,
    genre TEXT NOT NULL,
    console TEXT NOT NULL);
```

```
CREATE TABLE characters
    (charName TEXT NOT NULL,
    charGender TEXT NOT NULL,
    charClass INTEGER NOT NULL,
    charRelease INTEGER NOT NULL,
    title TEXT NOT NULL,
    FOREIGN KEY (title) REFERENCES games (title)
    );
```

```
CREATE TABLE items
    (itemName TEXT PRIMARY KEY NOT NULL,
    itemType TEXT NOT NULL,
    consumable TEXT NOT NULL,
    itemRelease INTEGER NOT NULL,
    title TEXT NOT NULL,
    FOREIGN KEY (title) REFERENCES games (title)
    );
```

3. Inserting data:

To insert data within a table, use the following sample code:

```
INSERT INTO table_name (column1, column2, column3, column4, column5) VALUES
    (value1, value2, value3, value4, value5);
INSERT INTO table_name (column1, column2, column3, column4, column5) VALUES
    (value6, value7, value8, value9, value10);
```

You can also insert data in bulk and one segment of code by adding a comma after the end of each set of data.

```
INSERT INTO table_name (column1, column2, column3, column4, column5) VALUES
    (value1, value2, value3, value4, value5),
    (value6, value7, value8, value9, value10);
```

When inserting data into a table, you must make sure that when you are listing your columns, they are in the same order that you created them. In addition, the values must also be in the same order.

Practice inserting data into the tables:

```
INSERT INTO games (title, studio, releaseYear, genre, console) VALUES
    ('Valorant', 'Riot', 2020, 'Tactical FPS', 'PC'),
    ('Genshin Impact', 'Hoyoverse', 2020, 'Open-world', 'Multiplatform'),
```

```
('Pokemon Arceus', 'Game Freak', 2022, 'Open-world RPG', 'Switch'),
('Fire Emblem: Three Houses', 'Intelligent Systems', 2019, 'Tactical RPG', 'Switch'),
('Overwatch', 'Blizzard', 2016, 'FPS', 'Multiplatform');
```

```
INSERT INTO characters (charName, charGender, charClass, charRelease, title) VALUES
('Yoru', 'Male', 'Duelist', 2022, 'Valorant'),
('Wanderer', 'Male', 'Catalyst', 2022, 'Genshin Impact'),
('Chamber', 'Male', 'Sentinel', 2021, 'Valorant'),
('Kiriko', 'Female', 'Support', 2022, 'Overwatch'),
('Widowmaker', 'Female', 'DPS', 2016, 'Overwatch');
```

```
INSERT INTO items (itemName, itemType, consumable, itemRelease, title) VALUES
('Vandal', 'Weapon', 'Yes', 2020, 'Valorant'),
('Tullaytullah’s Remembrance', 'Weapon', 'No', 2022, 'Genshin Impact'),
('Vivichoke', 'Food', 'Yes', 2022, 'Pokemon Arceus'),
('Oran Berry', 'Food', 'Yes', 2022, 'Pokemon Arceus'),
('Weapon charm', 'Cosmetic', 'No', 2016, 'Overwatch');
```

4. Querying data:

There are multiple ways to access data within your database. To see what tables are in your database, use this code:

```
.tables
```

To display *all* columns within your table, type:

```
SELECT * FROM table_name;
```

“*” is used when querying *all* columns within a table.

To query your table, ordered by certain columns, you will specify the column name and how you would like it to be listed: ascending (ASC) or descending (DESC) or random.

```
SELECT * FROM table_name ORDER BY column2 ASC/DESC;

SELECT * FROM table_name ORDER BY RANDOM();
```

To query your table, but only certain columns or values, you will specify the column name and the parameters by which you want them listed. This is especially important for listing your integer values. These are a couple different examples:

```
SELECT * FROM table_name WHERE column3=‘Yes’;

SELECT * FROM table_name WHERE column3 BETWEEN 1 AND 4;
```

Practice querying the following lines of code:

```
SELECT * FROM games;

SELECT * FROM characters ORDER BY releaseYear DESC;

SELECT * FROM items ORDER BY RANDOM ();

SELECT * FROM items WHERE consumable='Yes';

SELECT * FROM characters WHERE charRelease BETWEEN 2021 AND 2022 ORDER BY charRelease ASC;
```

5. Saving and ending case:

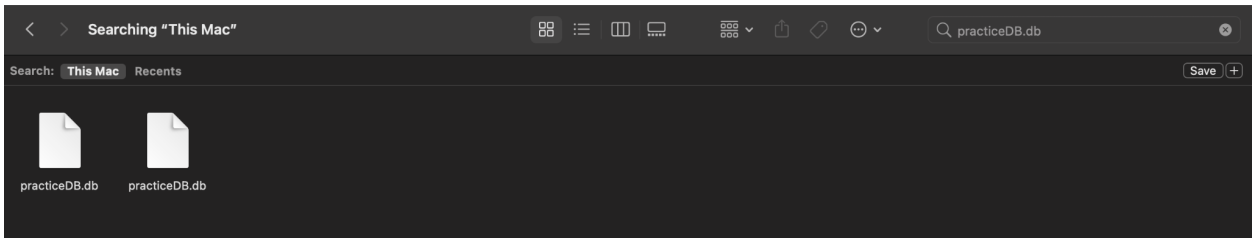
To quit out of Sqlite use the following line of code:

```
.quit
```

When you log in again, to access the previous database, use the ".open" command.

```
.open practiceDB.db
```

Once created, the database will be saved to your mac device. To find and locate the database file, simply go to your finder and type in the database's name and click on "This Mac". In this practice, you would search "practiceDB.db".



Code Case #2 Practice:

1. Create the following tables:

```
CREATE TABLE users (
        userID INTEGER PRIMARY KEY NOT NULL,
        firstName TEXT NOT NULL,
        lastName TEXT NOT NULL,
        age INTEGER NOT NULL,
```

```
    email TEXT NOT NULL
    );
```

```
CREATE TABLE products (
    productID INTEGER PRIMARY KEY NOT NULL,
    productName TEXT NOT NULL,
    price REAL NOT NULL,
    stockQuantity INTEGER NOT NULL,
    category TEXT NOT NULL
    );
```

```
CREATE TABLE orders (
    orderID INTEGER PRIMARY KEY NOT NULL,
    userID INTEGER NOT NULL,
    orderDate DATE NOT NULL,
    totalAmount REAL NOT NULL,
    orderAddress TEXT NOT NULL,
    FOREIGN KEY (userID) REFERENCES users (userID)
    );
```

2. Insert the following data into the tables:

```
INSERT INTO users (userID, firstName, lastName, age, email) VALUES
    (1, 'John', 'Micheal', 30, 'john.micheal@example.com'),
    (2, 'Jenette', 'Smith', 25, 'jen.smi@example.com'),
    (3, 'Brandon', 'Johnson', 35, 'johnson2brandon@example.com'),
    (4, 'Avery', 'Williams', 28, 'avery2757@example.com'),
    (5, 'Charlie', 'Bowing', 40, 'c.bowing1345@example.com');
```

```
INSERT INTO products (productID, productName, price, stockQuantity, category) VALUES
    (1, 'Laptop', 1200.00, 50, 'Electronics'),
    (2, 'Smartphone', 699.99, 100, 'Electronics'),
    (3, 'Bookshelf', 149.99, 30, 'Furniture'),
    (4, 'Coffee Maker', 49.95, 75, 'Appliances'),
    (5, 'Running Shoes', 79.99, 50, 'Clothing');
```

```
INSERT INTO orders (userID, orderDate, totalAmount, orderAddress) VALUES
    (1, '2023-01-01', 2500.00, '123 Maine St.'),
    (2, '2023-02-15', 500.00, '3140 Waialae Ave.'),
    (3, '2023-03-10', 120.00, '3140 Waialae Ave.'),
    (4, '2023-04-05', 700.00, '456 Jefferson Rd.'),
    (5, '2023-05-20', 300.00, '808 Rainbow Rd.');
```

3. Query the following data from the tables:

```
SELECT * FROM orders;

SELECT * FROM users ORDER BY age DESC;

SELECT * FROM products ORDER BY RANDOM ();

SELECT * FROM users WHERE firstName='John';

SELECT * FROM orders WHERE totalAmount BETWEEN 300.00 AND 700.00 ORDER BY
userID ASC;
```

Useful Links

Sqlite: https://www.sqlite.org/index.html

How to install SQLite on Mac, Windows, and Linux:

https://www.servermania.com/kb/articles/install-sqlite

SQLite features: https://www.sqlite.org/features.html

SQLite downloads: https://www.sqlite.org/download.html

Sqlite datatypes: https://www.w3resource.com/sqlite/sqlite-data-types.php

How to create a database, create tables, insert data, and query results:

https://www.youtube.com/watch?v=HQKwgk6XkIA&t=625s&ab_channel=Avery

Summary

Overall SQLite is a versatile and reliable database engine that is well suited for applications with moderate data storage and access requirements. Its simplicity, portability and performance make it a popular choice in various development environments. The use for SQLite and MySQL depends on the requirements for the project at hand. SQLite is favored for its simplicity and for smaller projects. While SQL is chosen for larger applications with higher

scalability needs and a client-server architecture. Despite this, it remains a strong competitor among other database programs today.

References

Hipp, R. D. (2023, November 23). *SQLite*. Retrieved from https://www.sqlite.org/index.html