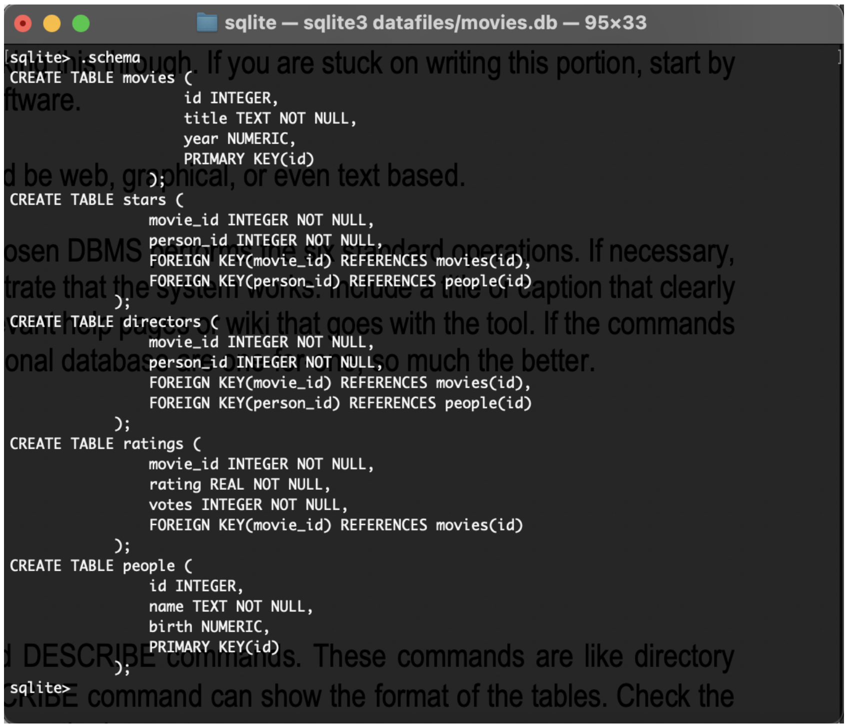
Software Discovery

SQLite will be the database management system we will use. SQLite is a simple, reliable, and serverless DBMS and "one of the most widely used database systems in the world"[1]. It is primarily used in mobile, laptops, and embedded systems that have limited memory space. It has all the basic features other DBMSs have, plus more. SQLite can be used with the Command Line Terminal or a Graphical User Interface. The best thing about SQLite is that it does not need a separate server to function since it is a file-based DBMS.

We chose SQLite because it is simple and easy to set up for our project. One drawback of SQLite is that it can not handle large-scale data, but for our project purposes, this is not a problem. The SQLite Pragma table_info(<table name>) does not show the foreign keys column.

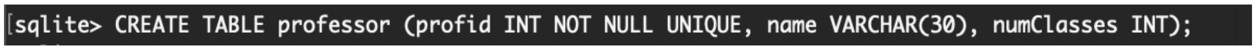
The work area of SQLite:



sqlite — sqlite3 datafiles/movies.db — 95×33

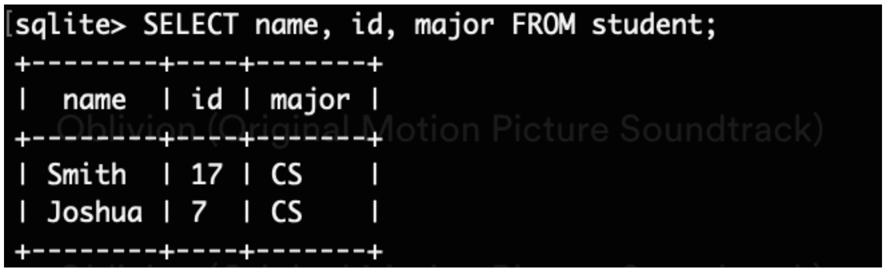
```
sqlite> .schema
CREATE TABLE movies (
                    id INTEGER,
                    title TEXT NOT NULL,
                    year NUMERIC,
                    PRIMARY KEY(id)
                );
CREATE TABLE stars (
                movie_id INTEGER NOT NULL,
                person_id INTEGER NOT NULL,
                FOREIGN KEY(movie_id) REFERENCES movies(id),
                FOREIGN KEY(person_id) REFERENCES people(id)
            );
CREATE TABLE directors (
                movie_id INTEGER NOT NULL,
                person_id INTEGER NOT NULL,
                FOREIGN KEY(movie_id) REFERENCES movies(id),
                FOREIGN KEY(person_id) REFERENCES people(id)
            );
CREATE TABLE ratings (
                movie_id INTEGER NOT NULL,
                rating REAL NOT NULL,
                votes INTEGER NOT NULL,
                FOREIGN KEY(movie_id) REFERENCES movies(id)
            );
CREATE TABLE people (
                id INTEGER,
                name TEXT NOT NULL,
                birth NUMERIC,
                PRIMARY KEY(id)
            );
sqlite>
```

CREATE TABLE script: SQLite will create the specified table by including the name and columns



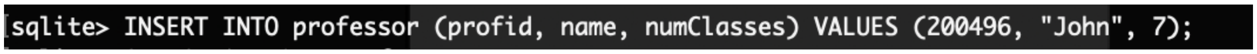
```
sqlite> CREATE TABLE professor (profid INT NOT NULL UNIQUE, name VARCHAR(30), numClasses INT);
```

SELECT script: SQLite will display the specified columns of a table



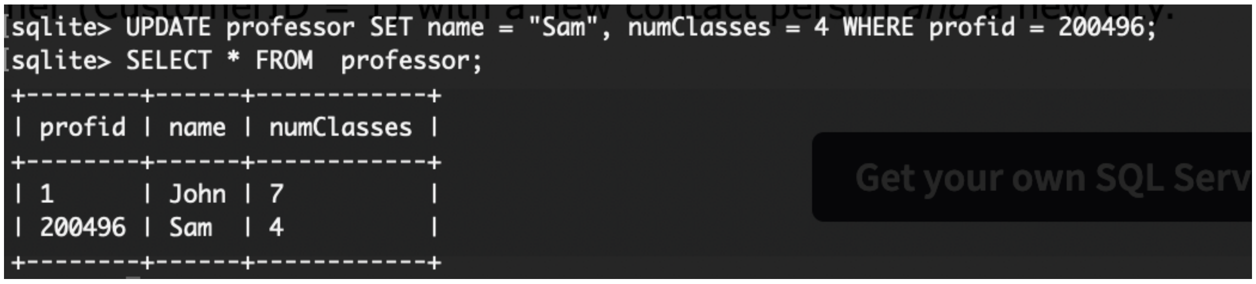
```
sqlite> SELECT name, id, major FROM student;
+--------+----+-------+
|  name  | id | major |
+--------+----+-------+
| Smith  | 17 | CS    |
| Joshua | 7  | CS    |
+--------+----+-------+
```

INSERT script: SQLite can insert values to specific columns in a table



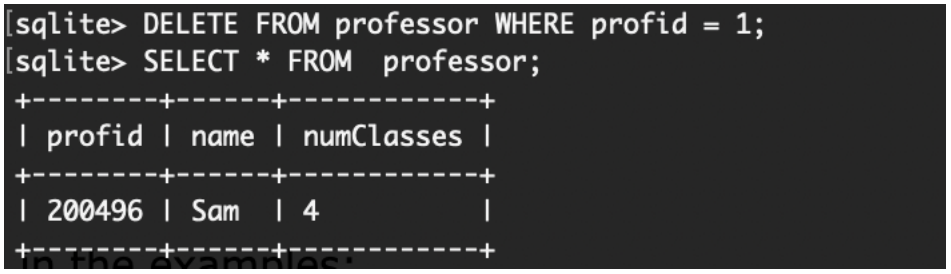
```
sqlite> INSERT INTO professor (profid, name, numClasses) VALUES (200496, "John", 7);
```

UPDATE script: SQLite will update the columns of the specified record



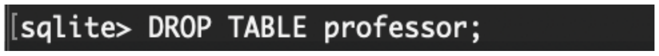
```
sqlite> UPDATE professor SET name = "Sam", numClasses = 4 WHERE profid = 200496;
sqlite> SELECT * FROM  professor;
+--------+------+------------+
| profid | name | numClasses |
+--------+------+------------+
| 1      | John | 7          |
| 200496 | Sam  | 4          |
+--------+------+------------+
```

DELETE script: SQLite will delete the records of a specified table



```
sqlite> DELETE FROM professor WHERE profid = 1;
sqlite> SELECT * FROM  professor;
+--------+------+------------+
| profid | name | numClasses |
+--------+------+------------+
| 200496 | Sam  | 4          |
+--------+------+------------+
```

DROP TABLE script: SQLite will delete the table and all the information contained in it.



```
sqlite> DROP TABLE professor;
```

REFERENCES

[1] D. R. Hipp, SQLite. “About SQLite”, [Online]. (2023). Available: https://www.sqlite.org/about.html