

Bocchi-sensei! Teach me SQLite

*by
Okabe*

Contents

Overview of SQLite	1
The REPL	1
Additional Meta-Commands	1
Python Library	2
Vicky	2
Additional Information	2
Author's Notes	3
Appendix	3

Bocchi-sensei! Teach me SQLite

by
Okabe

Overview of SQLite

SQLite has both, as will be used here, a REPL and library for interacting with SQL databases. This database engine is not networked like others, but is just a file on your system. This can be preferable to *ad hoc* or small applications.

The REPL

or command line version of SQLite.

The REPL -- inside of a database -- can be accessed by running:

```
sqlite3 path/to/db
```

The file path can be omitted if you would prefer an 'in-memory' database. Once inside of the database, you can run various SQL queries *e.g.*:

```
sqlite3 files/quotes.db
```

```
CREATE TABLE quotes (addedby text, name text, quote text);
```

```
INSERT INTO quotes VALUES("okabe", "f0urf0ur", "how sqlite");
```

Additional Meta-Commands

Command	Action
<code>.open path</code>	Opens a database
<code>.databases</code>	Lists open databases
<code>.tables</code>	Lists tables
<code>.schema</code>	CREATE command history
<code>.save path</code>	Saves an in memory database
<code>.clone path</code>	Clones database
<code>.quit, .exit and ^d</code>	Closes the REPL

Python Library

The SQLite library in python can be accessed by importing the sqlite3 library.

```
import sqlite3
```

Before you can interact with the database, you need to make a connection and cursor object.

```
conn = sqlite3.connect(/path/to/db)
```

```
cur = conn.cursor()
```

Now that you've made a cursor you can run SQL queries. Note the ';' has been omitted.

```
cur.execute("INSERT INTO quotes VALUES('okabe', 'okabe', 'i  
fucking hate lua')
```

```
# Only after changing the database
```

```
conn.commit()
```

A optional secondary argument to `cur.execute` is a tuple used for substitutions similar to `printf`. **Do not use formatting, use the tuples!**

And lastly closing the database with `conn.close()`.

Vicky

In Vicky, SQL operations are abstracted by `lib.sql`. The aforementioned routine is simplified by the two following functions

- `list_from_db` – Intended to list from db -- hence the name -- and takes the filepath, query and tuple used for replacement.
- `add_to_db` – Takes the same arguments as `list_from_db`. Is used to modify the database.

Additional Information

You can get some more information at:

- <https://sqlite.org> – The SQLite homepage.
- `man sqlite3` – The SQLite3 REPL manual page.
- <https://docs.python.org/3/library/sqlite3.html> – Python SQLite3 Documentation.

Author's Notes

Damn I really suck at documentation and writing in general. You should take everything I say with a grain of salt because I'm basically talking out of my ass and suck at programming, as evidenced by the shit module I wrote. I wrote this in Groff's mom, in case you wanted to know. There was going to be a part about the C API. Also, when I said in irc "<", I meant "<<<".

Fuck this is cringe and the title is weeb shit.

-Okabe

Appendix

A) Running `sqlite3` with the `"-init FILE"` flag can be used to read from a schema. This can be used instead of `"sqlite3 path/db < schema.sql"`.