
sqlitebiter Documentation

Release 0.23.1

Tsuyoshi Hombashi

Oct 08, 2018

Table of Contents

1	sqlitebiter	1
1.1	Summary	1
1.2	Features	1
2	Installation	3
2.1	Install via pip (recommended)	3
2.2	Installation for Debian/Ubuntu from a deb package	3
2.3	Installing executable files in Windows	3
2.4	Installation for macOS via Homebrew	4
2.5	Command Completion	4
3	Dependencies	5
3.1	Python package dependencies	5
3.1.1	Mandatory dependencies	5
3.1.2	Google Sheets dependencies (Optional)	5
3.1.3	Test dependencies	6
3.1.4	Misc dependencies (Optional)	6
3.2	Dependencies other than Python packages (Optional)	6
4	Usage	7
4.1	sqlitebiter command help	7
4.2	Create a SQLite database from CSV/Excel/JSON/SQLite/etc. files	8
4.2.1	Examples	8
4.2.2	sqlitebiter file subcommand help	9
4.2.3	Supported data formats	9
4.2.4	Table naming convention	10
4.3	Create a SQLite database from URL	11
4.3.1	Example	11
4.3.2	sqlitebiter url subcommand help	13
4.4	Create a SQLite database from Google Sheets	14
4.4.1	Requirements	14
4.4.2	Example	14
4.4.3	sqlitebiter gs subcommand help	14
5	Indices and tables	17
6	Links	19

1.1 Summary

A CLI tool to convert CSV / Excel / HTML / JSON / Jupyter Notebook / LDJSON / LTSV / Markdown / SQLite / SSV / TSV / Google-Sheets to a SQLite database file.

1.2 Features

- **Create a SQLite database file from:**
 - **File(s):**
 - * CSV / Tab separated values (TSV) / Space separated values (SSV)
 - * Microsoft Excel TM
 - * HTML
 - * JSON
 - * Jupyter Notebook
 - * Labeled Tab-separated Values (LTSV)
 - * Line-delimited JSON(LDJSON) / NDJSON / JSON Lines
 - * Markdown
 - * Mediawiki
 - * SQLite
 - Google Sheets

- URL (scrape tabular data from web pages)
- Multi-byte character support
- Automatic file encoding detection

2.1 Install via pip (recommended)

sqlitebiter can be installed via `pip` (Python package manager).

Example

```
pip install sqlitebiter
```

2.2 Installation for Debian/Ubuntu from a deb package

1. `wget https://github.com/thombashi/sqlitebiter/releases/download/<version>/sqlitebiter-<version>-amd64.deb`
2. `dpkg -iv sqlitebiter-<version>-amd64.deb`

Example

```
$ wget https://github.com/thombashi/sqlitebiter/releases/download/v0.20.0/sqlitebiter_0.20.0_amd64.deb
$ sudo dpkg -i sqlitebiter_0.20.0_amd64.deb
```

2.3 Installing executable files in Windows

sqlitebiter can be used in Windows environments without Python installation as follows:

1. Navigate to <https://github.com/thombashi/sqlitebiter/releases>
2. Download the latest version of the `sqlitebiter_win_x64.zip`
3. Unzip the file

4. Execute `sqlitebiter.exe` in either Command Prompt or PowerShell

```
>cd sqlitebiter_win_x64
>sqlitebiter.exe -h
Usage: sqlitebiter [OPTIONS] COMMAND [ARGS]...

Options:
  --version            Show the version and exit.
  -o, --output-path PATH Output path of the SQLite database file. Defaults to
                        'out.sqlite'.
  -a, --append         append table(s) to existing database.
  -i, --index TEXT     comma separated attribute names to create indices.
  --replace-symbol TEXT Replace symbols in attributes.
  -v, --verbose
  --debug              for debug print.
  -q, --quiet          suppress execution log messages.
  -h, --help          Show this message and exit.

Commands:
  completion A helper command to setup command completion.
  configure  Configure the following application settings: (1) Default...
  file       Convert tabular data within CSV/Excel/HTML/JSON/Jupyter...
  gs         Convert a spreadsheet in Google Sheets to a SQLite database...
  url        Scrape tabular data from a URL and convert data to a SQLite...
```

2.4 Installation for macOS via Homebrew

```
$ brew tap thombashi/sqlitebiter
$ brew install sqlitebiter
```

- [Homebrew Formula](#)

2.5 Command Completion

```
To setup for bash:

  sqlitebiter completion bash >> ~/.bashrc

To setup for zsh:

  sqlitebiter completion zsh >> ~/.zshrc
```


Python 2.7+ or 3.4+

Python package dependencies are as follows.

3.1 Python package dependencies

3.1.1 Mandatory dependencies

Following mandatory Python packages are automatically installed during `sqlitebiter` installation process:

- `appconfigpy`
- `click`
- `colorama`
- `logbook`
- `msgfy`
- `nbformat`
- `path.py`
- `pathvalidate`
- `pytablereader`
- `SimpleSQLite`
- `typepy`

3.1.2 Google Sheets dependencies (Optional)

Following Python packages are required to [manual installation](#) when you use Google Sheets feature:

- gspread
- oauth2client
- pyOpenSSL

The above packages can be installed with the following pip command;

```
$ pip install sqlitebiter[gs]
```

3.1.3 Test dependencies

- pytest
- pytest-runner
- responses
- sqliteschema
- tox

3.1.4 Misc dependencies (Optional)

- lxml
- **py pandoc**
 - required when converting MediaWiki files

3.2 Dependencies other than Python packages (Optional)

- libxml2 (faster HTML/Markdown conversion)
- pandoc (required when converting MediaWiki files)

4.1 sqlitebiter command help

sqlitebiter has following subcommands:

- **file:** Convert tabular data within CSV/Excel/HTML/JSON/LTSV/Markdown/SQLite/TSV file(s) to a SQLite database file.
 - *Create a SQLite database from CSV/Excel/JSON/SQLite/etc. files*
- **url:** Scrape tabular data from a URL and convert data to a SQLite database file.
 - *Create a SQLite database from URL*
- **gs:** Convert a spreadsheet in Google Sheets to a SQLite database file.
 - *Create a SQLite database from Google Sheets*
- **configure:** Configure the application settings

```
Usage: sqlitebiter [OPTIONS] COMMAND [ARGS]...

Options:
  --version                Show the version and exit.
  -o, --output-path PATH  Output path of the SQLite database file. Defaults to
                          'out.sqlite'.
  -a, --append            append table(s) to existing database.
  -i, --index TEXT       comma separated attribute names to create indices.
  --replace-symbol TEXT  Replace symbols in attributes.
  -v, --verbose          for debug print.
  --debug                suppress execution log messages.
  -q, --quiet            Show this message and exit.
  -h, --help

Commands:
  completion  A helper command to setup command completion.
  configure   Configure the following application settings: (1) Default...
```

(continues on next page)

(continued from previous page)

file	Convert tabular data within CSV/Excel/HTML/JSON/Jupyter...
gs	Convert a spreadsheet in Google Sheets to a SQLite database...
url	Scrape tabular data from a URL and convert data to a SQLite...

4.2 Create a SQLite database from CSV/Excel/JSON/SQLite/etc. files

sqlitebiter file is a subcommand to convert tabular data file(s) to a SQLite database file.

4.2.1 Examples

Using wildcard to convert multiple files. File formats are automatically detected from the extensions.

Example

```
$ ls
sample_data.csv  sample_data.xlsx  sample_data_multi.json  sample_data_
↳single.json
$ sqlitebiter -o sample.sqlite file *
[INFO] sqlitebiter file: convert 'sample_data.csv' to 'sample_data' table
[INFO] sqlitebiter file: convert 'sample_data_multi.json' to 'table_a'
↳table
[INFO] sqlitebiter file: convert 'sample_data_multi.json' to 'table_b'
↳table
[INFO] sqlitebiter file: convert 'sample_data_single.json' to 'sample_
↳data_single' table
[INFO] sqlitebiter file: convert 'sample_data.xlsx' to 'samplesheet1'
↳table
[INFO] sqlitebiter file: convert 'sample_data.xlsx' to 'samplesheet3'
↳table
[INFO] sqlitebiter file: converted results: source=4, success=6, created-
↳table=6
[INFO] sqlitebiter file: database path: sample.sqlite
```

Output

```
$ sqlite3 sample.sqlite .schema
CREATE TABLE IF NOT EXISTS '_source_info_' ("source_id" INTEGER NOT NULL,
↳ "dir_name" TEXT, "base_name" TEXT NOT NULL, "format_name" TEXT NOT
↳NULL, "dst_table" TEXT NOT NULL, size INTEGER, mtime INTEGER);
CREATE TABLE IF NOT EXISTS 'sample_data' ("attr_a" INTEGER, "attr_b"
↳REAL, "attr_c" TEXT);
CREATE TABLE IF NOT EXISTS 'table_a' ("attr_a" INTEGER, "attr_b" REAL,
↳"attr_c" TEXT);
CREATE TABLE IF NOT EXISTS 'table_b' (a INTEGER, b REAL);
CREATE TABLE IF NOT EXISTS 'sample_data_single' ("attr_a" INTEGER, "attr_
↳b" REAL, "attr_c" TEXT);
CREATE TABLE IF NOT EXISTS 'samplesheet1' (a INTEGER, b REAL, c TEXT);
CREATE TABLE IF NOT EXISTS 'samplesheet3' (aa INTEGER, ab TEXT, ac TEXT);
```

Designate multiple file path to convert:

Example

```
$ sqlitebiter file sample_data.csv sample_data.xlsx
[INFO] sqlitebiter file: convert 'sample_data.csv' to 'sample_data' table
[INFO] sqlitebiter file: convert 'sample_data.xlsx' to 'samplesheet1'
↪table
[INFO] sqlitebiter file: convert 'sample_data.xlsx' to 'samplesheet3'
↪table
[INFO] sqlitebiter file: converted results: source=2, success=3, created-
↪table=3
[INFO] sqlitebiter file: database path: out.sqlite
```

Note:

- Available JSON Schema is limited. Acceptable format described in [here](#)
- Wildcard characters cannot use in Windows environments

You could specify converting file format with the `--format` option

Example

```
$ sqlitebiter file --format csv sample_data
[INFO] sqlitebiter file: convert 'sample_data.csv' to 'sample_data' table
[INFO] sqlitebiter file: converted results: source=1, success=1, created-
↪table=1
[INFO] sqlitebiter file: database path: out.sqlite
```

4.2.2 sqlitebiter file subcommand help

Usage: sqlitebiter file [OPTIONS] [FILES]...

Convert tabular data within CSV/Excel/HTML/JSON/Jupyter Notebook/LDJSON/LTSV/Markdown/Mediawiki/SQLite/SSV/TSV file(s) **or** named pipes to a SQLite database file.

Options:

<code>-r, --recursive</code>	Read all files under each directory, recursively.
<code>--pattern PATTERN</code>	Convert files matching PATTERN.
<code>--exclude PATTERN</code>	Exclude files matching PATTERN.
<code>--follow-symlinks</code>	Follow symlinks.
<code>-f, --format [csv excel html json json_↪lines jsonl ldjson ltsv markdown mediawiki ndjson sqlite ssv tsv ipynb]</code>	Data format to loading (auto-detect from file extensions in default).
<code>--encoding ENCODING</code>	Encoding to load files. Auto-detection from files in default.
<code>-h, --help</code>	Show this message and exit.

Documentation: <https://sqlitebiter.rtfid.io/> Issue tracker: <https://github.com/thombashi/sqlitebiter/issues>

4.2.3 Supported data formats

Following table shows that the supported data formats:

Table 1: Available data formats

Format	File Extension	Remarks
CSV	.csv	
Excel	.xlsx/.xls	Create table for each sheet in the Excel workbook.
HTML	.html/.htm	Scrape tabular data from <table> tags in the HTML file. And create table for each <table> tag data.
JSON	.json	
Jupyter Notebook	.ipynb	
Line-delimited JSON	.jsonl/.ldjson/.ndjson	
LTSV	.ltsv	
Markdown	.md	Extract tabular data in the Markdown file. And create a table for each <table> tabular data.
SQLite	.sqlite/.sqlite3	
TSV	.tsv	

4.2.4 Table naming convention

Table name automatically decided as follows for each format:

Format	Table Name
CSV	<filename>
Excel	<Sheet name>
HTML	<title>_<key>. <title> replaced with the title tag of the page. <key> replaced with: (1) id attribute of the table tag. (2) unique string if id attribute not present in the table tag.
JSON	
LTSV	<filename>
Markdown	<filename>
TSV	<filename>

- **Common behavior**

- **<filename> replaced with filename of converting file (without extensions)**

- * e.g. If the input file name is sample.csv, <filename> is sample

If a created table name already exists in the database, the behavior differs depending on the existing table (after this referred to as A) and create table (after this referred to as B) structure:

1. **A and B has the same table name and table structure**
 - Append creating table data to the existing table data
2. **A and B has the same table name, but different table structure**
 - sqlitebiter try to create unique table name for B by appending suffix id number

4.3 Create a SQLite database from URL

sqlitebiter url is a subcommand to fetch table data from the Internet and convert to a SQLite database file.

4.3.1 Example

Following is an example that converts HTML table tags within a web page to SQLite tables by the web page URL.

Example

```
$ sqlitebiter url "https://en.wikipedia.org/wiki/Comparison_of_firewalls"
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html1' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html2' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html3' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html4' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html5' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html6' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html7' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html8' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html9' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html10' table
[INFO] sqlitebiter url: convert 'Comparison_of_firewalls' to 'Comparison_
↳of_firewalls_Wikipedia_html11' table
[INFO] sqlitebiter url: converted results: source=1, success=11, created-
↳table=11
[INFO] sqlitebiter url: database path: out.sqlite
```

Output

```
$ sqlite3 out.sqlite .schema
CREATE TABLE IF NOT EXISTS '_source_info_' ("source_id" INTEGER NOT NULL,
↳ "dir_name" TEXT, "base_name" TEXT NOT NULL, "format_name" TEXT NOT_
↳ NULL, "dst_table" TEXT NOT NULL, size INTEGER, mtime INTEGER);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html1'_
↳ (Firewall TEXT, License TEXT, [Cost and usage limits] TEXT, OS TEXT);
```

(continues on next page)

(continued from previous page)

```
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html2'
↳(Firewall TEXT, License TEXT, Cost TEXT, OS TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html3'
↳([Can Target:] TEXT, [Changing default policy to accept/reject (by
↳issuing a single rule)] TEXT, [IP destination address(es)] TEXT, [IP
↳source address(es)] TEXT, [TCP/UDP destination port(s)] TEXT, [TCP/UDP
↳source port(s)] TEXT, [Ethernet MAC destination address] TEXT,
↳[Ethernet MAC source address] TEXT, [Inbound firewall (ingress)] TEXT,
↳[Outbound firewall (egress)] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html4'
↳([Can:] TEXT, [work at OSI Layer 4 (stateful firewall)] TEXT, [work at
↳OSI Layer 7 (application inspection)] TEXT, [Change TTL? (Transparent
↳to traceroute)] TEXT, [Configure REJECT-with answer] TEXT, [DMZ (de-
↳militarized zone) - allows for single/several hosts not to be
↳firewalled.] TEXT, [Filter according to time of day] TEXT, [Redirect
↳TCP/UDP ports (port forwarding)] TEXT, [Redirect IP addresses
↳(forwarding)] TEXT, [Filter according to User Authorization] TEXT,
↳[Traffic rate-limit / QoS] TEXT, Tarpit TEXT, Log TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html5'
↳([Features:] TEXT, "Configuration: GUI_ text or both modes?" TEXT,
↳"Remote Access: Web (HTTP)_ Telnet_ SSH_ RDP_ Serial COM RS232_ ..."
↳TEXT, [Change rules without requiring restart?] TEXT, [Ability to
↳centrally manage all firewalls together] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html6'
↳([Features:] TEXT, [Modularity: supports third-party modules to extend
↳functionality?] TEXT, [IPS : Intrusion prevention system] TEXT, [Open-
↳Source License?] TEXT, [supports IPv6 ?] TEXT, [Class: Home /
↳Professional] TEXT, [Operating Systems on which it runs?] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html7'
↳([Can:] TEXT, "NAT44 (static_ dynamic w/o ports_ PAT)" TEXT, "NAT64_
↳NPTv6" TEXT, [IDS (Intrusion Detection System)] TEXT, [VPN (Virtual
↳Private Network)] TEXT, [AV (Anti-Virus)] TEXT, Sniffer TEXT,
↳[Profile selection] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html8'
↳([vteFirewall software] TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html9' (A
↳TEXT, B TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html10' (A
↳TEXT, B TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html11' (A
↳TEXT, B TEXT);
```

The attributes within the converted SQLite database may include symbols as the above. Symbols within attributes can be replaced by using `--replace-symbol` option. In the following example shows replace symbols to underscores.

Example

```
$ sqlitebiter --replace-symbol _ -q url "https://en.wikipedia.org/wiki/
↳Comparison_of_firewalls"
```

Output

```
$ sqlite3 out.sqlite .schema
CREATE TABLE IF NOT EXISTS '_source_info_' ("source_id" INTEGER NOT NULL,
↳ "dir_name" TEXT, "base_name" TEXT NOT NULL, "format_name" TEXT NOT
↳ NULL, "dst_table" TEXT NOT NULL, size INTEGER, mtime INTEGER);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html11'
↳(Firewall TEXT, License TEXT, "Cost_and_usage_limits" TEXT, OS TEXT);
```

(continues on next page)

(continued from previous page)

```

CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html2' (
↳(Firewall TEXT, License TEXT, Cost TEXT, OS TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html3' (
↳"Can_Target" TEXT, "Changing_default_policy_to_accept_reject_by_
↳issuing_a_single_rule" TEXT, "IP_destination_address_es" TEXT, "IP_
↳source_address_es" TEXT, "TCP_UDP_destination_port_s" TEXT, "TCP_UDP_
↳source_port_s" TEXT, "Ethernet_MAC_destination_address" TEXT,
↳"Ethernet_MAC_source_address" TEXT, "Inbound_firewall_ingress" TEXT,
↳"Outbound_firewall_egress" TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html4' (
↳(Can TEXT, "work_at_OSI_Layer_4_stateful_firewall" TEXT, "work_at_OSI_
↳Layer_7_application_inspection" TEXT, "Change_TTL_Transparent_to_
↳traceroute" TEXT, "Configure_REJECT_with_answer" TEXT, "DMZ_de_
↳militarized_zone_allows_for_single_several_hosts_not_to_be_firewalled"
↳TEXT, "Filter_according_to_time_of_day" TEXT, "Redirect_TCP_UDP_ports_
↳port_forwarding" TEXT, "Redirect_IP_addresses_forwarding" TEXT,
↳"Filter_according_to_User_Authorization" TEXT, "Traffic_rate_limit_QoS
↳" TEXT, Tarpit TEXT, Log TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html5' (
↳(Features TEXT, "Configuration_GUI_text_or_both_modes" TEXT, "Remote_
↳Access_Web_HTTP_Telnet_SSH_RDP_Serial_COM_RS232" TEXT, "Change_rules_
↳without_requiring_restart" TEXT, "Ability_to_centrally_manage_all_
↳firewalls_together" TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html6' (
↳(Features TEXT, "Modularity_supports_third_party_modules_to_extend_
↳functionality" TEXT, "IPS_Intrusion_prevention_system" TEXT, "Open_
↳Source_License" TEXT, "supports_IPv6" TEXT, "Class_Home_Professional"
↳TEXT, "Operating_Systems_on_which_it_runs" TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html7' (
↳(Can TEXT, "NAT44_static_dynamic_w_o_ports_PAT" TEXT, "NAT64_NPTv6"
↳TEXT, "IDS_Intrusion_Detection_System" TEXT, "VPN_Virtual_Private_
↳Network" TEXT, "AV_Anti_Virus" TEXT, Sniffer TEXT, "Profile_selection"
↳TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html8' (
↳"vteFirewall_software" TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html9' (A
↳TEXT, B TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html10' (A
↳TEXT, B TEXT);
CREATE TABLE IF NOT EXISTS 'Comparison_of_firewalls_Wikipedia_html11' (A
↳TEXT, B TEXT);

```

4.3.2 sqlitebiter url subcommand help

Usage: sqlitebiter url [OPTIONS] URL

Scrape tabular data **from a URL and** convert data to a SQLite database file.

Options:

```

-f, --format [csv|excel|html|json|json_
↳lines|jsonl|ldjson|ltsv|markdown|mediawiki|ndjson|sqlite|ssv|tsv|ipynb]
Data format to loading (defaults to html).
-e, --encoding ENCODING          HTML page read encoding. Defaults to utf-8.
-p, --proxy PROXY                Specify a proxy in the form
[user:passwd@]proxy.server:port.

```

(continues on next page)

(continued from previous page)

```
-h, --help          Show this message and exit.

Documentation: https://sqlitebiter.rtfid.io/ Issue tracker:
https://github.com/thombashi/sqlitebiter/issues
```

4.4 Create a SQLite database from Google Sheets

`sqlitebiter gs` is a subcommand to convert Google Sheets to a SQLite database file.

4.4.1 Requirements

Following python packages are required to use Google Sheets feature.

- `gspread`
- `oauth2client`
- `pyOpenSSL`

Dependency Python package installation:

```
$ sudo pip install gspread
$ sudo pip install oauth2client pyopenssl
```

4.4.2 Example

Example

```
$ sqlitebiter -o sample.sqlite gs credentials-xxxxxxxxxxxxx.json_
↪samplebook
[INFO] sqlitebiter gs: convert 'samplebook' to 'sheet3' table
[INFO] sqlitebiter gs: convert 'samplebook' to 'sheet1' table
[INFO] sqlitebiter gs: converted results: source=1, success=2, created-
↪table=2
[INFO] sqlitebiter gs: database path: sample.sqlite
```

Output

```
$ sqlite3 sample.sqlite .schema
CREATE TABLE sqlite_sequence(name,seq);
CREATE TABLE IF NOT EXISTS '_source_info_' ("source_id" INTEGER NOT NULL,
↪ "dir_name" TEXT, "base_name" TEXT NOT NULL, "format_name" TEXT NOT_
↪NULL, "dst_table" TEXT NOT NULL, size INTEGER, mtime INTEGER);
CREATE TABLE IF NOT EXISTS 'sheet3' (a INTEGER, b REAL, c TEXT);
CREATE TABLE IF NOT EXISTS 'sheet1' (a INTEGER, b REAL, c TEXT);
```

4.4.3 `sqlitebiter gs` subcommand help

```
Usage: sqlitebiter gs [OPTIONS] CREDENTIALS TITLE
```

```
Convert a spreadsheet in Google Sheets to a SQLite database file.
```

```
CREDENTIALS: OAuth2 Google credentials file. TITLE: Title of the Google  
Sheets to convert.
```

```
Options:
```

```
-h, --help Show this message and exit.
```

```
Documentation: https://sqlitebiter.rtf.d.io/ Issue tracker:  
https://github.com/thombashi/sqlitebiter/issues
```


CHAPTER 5

Indices and tables

- genindex

CHAPTER 6

Links

- [pip](#): A tool for installing python packages
- [GitHub repository](#)
- [Issue tracker](#)
- [PyPI](#)

CHAPTER 7

Indices and tables

- genindex