How To Use The SQLite Dump Command

**Summary**: in this tutorial, you will learn how to use the SQLite dump command to back up and restore a database.

SQLite project delivers the sqlite3 tool that allows you to interact with the SQLite database using a command-line program.

By using the sqlite3 tool, you can use the SQL statements to query or update data in the database. Also, you can use special commands, which are known as dot commands to perform various useful database operations.

One of these dot-commands is the  .dump command that gives you the ability to dump the entire database or tables into a text file.

Dump the entire database into a file using the SQLite dump command

The following command opens a new SQLite database connection to the chinook.db file.

C:\sqlite>sqlite3 c:/sqlite/chinook.db

SQLite version 3.13.0 2016-05-18 10:57:30

Enter ".help" for usage hints.

sqlite>

Code language: JavaScript (javascript)

To dump a database into a file, you use the .dump command. The .dump command converts the entire structure and data of an SQLite database into a single text file.

By default, the .dump command outputs the SQL statements on screen. To issue the output to a file, you use the .output FILENAME command.

The following commands specify the output of the dump file to chinook.sql and dump the chinook database into the chinook.sql file.

sqlite> .output c:/sqlite/chinook.sql

sqlite> .dump

sqlite> .exit

Code language: JavaScript (javascript)

Dump a specific table using the SQLite dump command

To dump a specific table, you specify the table name after the .dump command. For example, the following command saves the albums table to the albums.sql file.

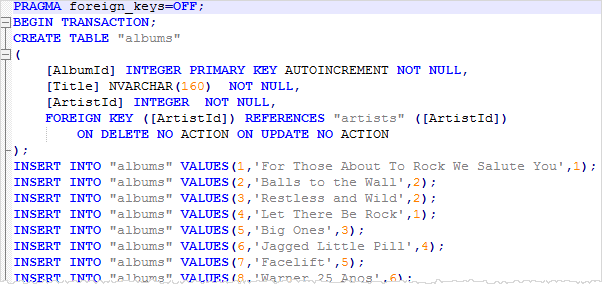
sqlite> .output c:/sqlite/albums.sql

sqlite> .dump albums

sqlite> .quit

Code language: JavaScript (javascript)

The following picture shows the contents of the albums.sql file.



Dump tables structure only using schema command

To dump the table structures in a database, you use the .schema command.

The following commands set the output file to chinook\_structure.sql file and save the table structures into the chinook\_structure.sql file:

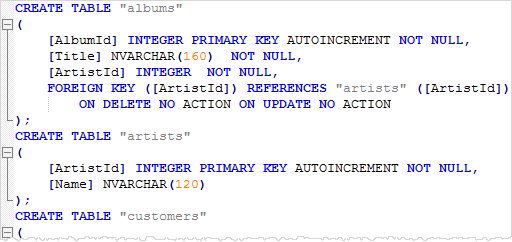
sqlite> .output c:/sqlite/chinook\_structure.sql

sqlite> .schema

sqlite> .quit

Code language: JavaScript (javascript)

The following picture shows the content of the chinook\_structure.sql file.



Dump data of one or more tables into a file

To dump the data of a table into a text file, you use these steps:

First, set the mode to insert using the .mode command as follows:

sqlite> .mode insert

Code language: CSS (css)

From now on, every [SELECT statement](https://www.sqlitetutorial.net/sqlite-select/) will issue the result as the [INSERT statements](https://www.sqlitetutorial.net/sqlite-insert/) instead of pure text data.

Second, set the output to a text file instead of the default standard output. The following command sets the output file to the data.sql file.

sqlite> .output data.sql

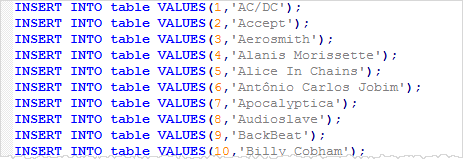
Code language: CSS (css)

Third, issue the [SELECT](https://www.sqlitetutorial.net/sqlite-select/) statements to query data from a table that you want to dump. The following command returns data from the artists table.

sqlite> select \* from artists;

Code language: SQL (Structured Query Language) (sql)

Check the content of the data.sql file, if everything is fine, you will see the following output:



To dump data from other tables, you need to issue the SELECT statements to query data from those tables.

Import a CSV File Into an SQLite Table

**Summary**: in this tutorial, you will learn various ways to import CSV data into an SQLite table using sqlite3 and SQLite Studio tools.

Importing a CSV file into a table using sqlite3 tool

In the first scenario, you want to import data from CSV file into a table that does not exist in the SQLite database.

1. First, the sqlite3 tool creates the table. The sqlite3 tool uses the first row of the CSV file as the names of the columns of the table.
2. Second, the sqlite3 tool import data from the second row of the CSV file into the table.

We will import a CSV file named city.csv with two columns: name and population. You can download it here for practicing.

[Download the city.csv file](https://www.sqlitetutorial.net/wp-content/uploads/2016/05/city.csv)

To import the c:\sqlite\city.csv file into the cities table:

First, set the mode to CSV to instruct the command-line shell program to interpret the input file as a CSV file. To do this, you use the .mode command as follows:

sqlite> .mode csv

Second, use the command .import FILE TABLE to import the data from the city.csv file into the cities table.

sqlite>.import c:/sqlite/city.csv cities

To verify the import, you use the command .schema to display the structure of the cities table.

sqlite> .schema cities

CREATE TABLE cities(

  "name" TEXT,

  "population" TEXT

);

Code language: SQL (Structured Query Language) (sql)

To view the data of the cities table, you use the following SELECT statement.

SELECT

name,

population

FROM

cities;

Code language: SQL (Structured Query Language) (sql)

In the second scenario, the table is already available in the database and you just need to import the data.

First, [drop](https://www.sqlitetutorial.net/sqlite-drop-table/) the cities table that you have created.

DROP TABLE IF EXISTS cities;

Code language: SQL (Structured Query Language) (sql)

Second, use the following [CREATE TABLE](https://www.sqlitetutorial.net/sqlite-create-table/) statement to create the table cities.

CREATE TABLE cities(

name TEXT NOT NULL,

population INTEGER NOT NULL

);

Code language: SQL (Structured Query Language) (sql)

If the table already exists, the sqlite3 tool uses all the rows, including the first row, in the CSV file as the actual data to import. Therefore, you should delete the first row of the CSV file.

The following commands import the city\_without\_header.csv file into the cities table.

sqlite> .mode csv

sqlite> .import c:/sqlite/city\_no\_header.csv cities

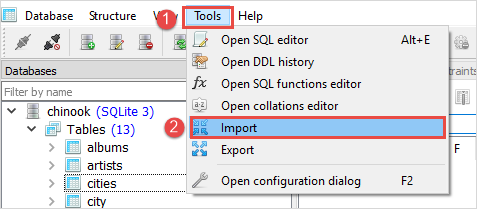
Code language: SQL (Structured Query Language) (sql)

Import a CSV file into a table using SQLite Studio

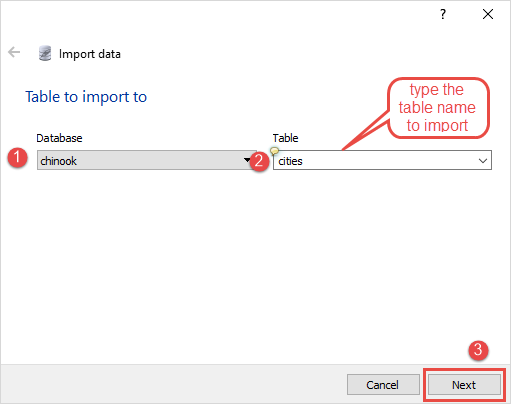
Most SQLite GUI tools provide the import function that allows you to import data from a file in CSV format, tab-delimited format, etc., into a table.

We will use the SQLite Studio to show you how to import a CSV file into a table with the assumption that the target table already exists in the database.

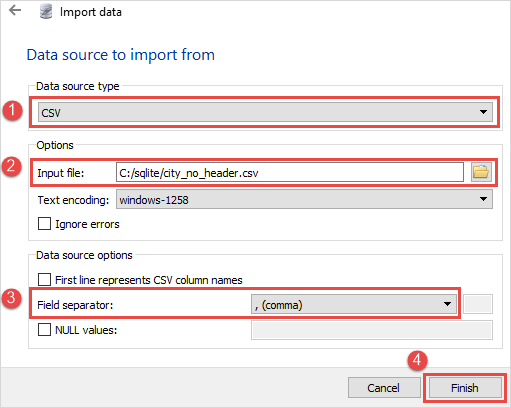
First, from the menu choose tool menu item.



Second, choose the database and table that you want to import data then click the **Next** button.



Third, choose CSV as the data source type, choose the CSV file in the **Input file** field, and choose the **,(comma)** option as the **Field separator** as shown in the picture below. Then click the **Finish** button to import the data.



# Export SQLite Database To a CSV File

**Summary**: in this tutorial, you will learn how to export SQLite database to a CSV file.

There are several ways to dump data from an SQLite database to a CSV file.

## Export SQLite Database to a CSV file using sqlite3 tool

SQLite project provides you with a command-line program called sqlite3 or sqlite3.exe on Windows. By using the sqlite3 tool, you can use the SQL statements and dot-commands to interact with the SQLite database.

To export data from the SQLite database to a CSV file, you use these steps:

1. Turn on the header of the result set using the .header on command.
2. Set the output mode to CSV to instruct the sqlite3 tool to issue the result in the CSV mode.
3. Send the output to a CSV file.
4. Issue the query to select data from the table to which you want to export.

The following commands select data from the customers table and export it to the data.csv file.

>sqlite3 c:/sqlite/chinook.db

sqlite> .headers on

sqlite> .mode csv

sqlite> .output data.csv

sqlite> SELECT customerid,

...> firstname,

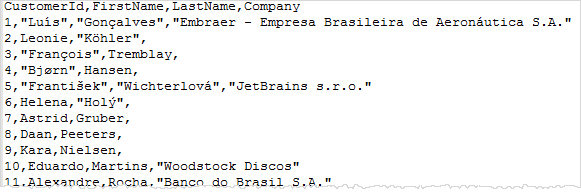
...> lastname,

...> company

...> FROM customers;

sqlite> .quit

If you check the data.csv file, you will see the following output.

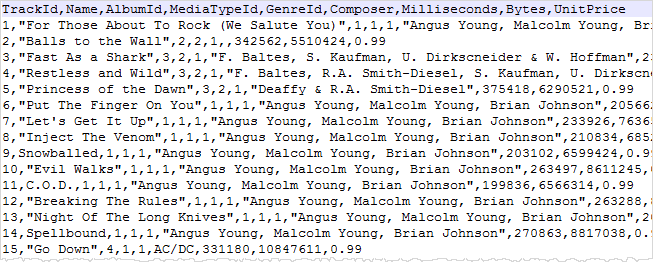


Besides using the dot-commands, you can use the options of the sqlite3 tool to export data from the SQLite database to a CSV file.

For example, the following command exports the data from the tracks table to a CSV file named tracks.csv.

>sqlite3 -header -csv c:/sqlite/chinook.db "select \* from tracks;" > tracks.csv

Code language: SQL (Structured Query Language) (sql)



If you have a file named query.sql that contains the script to query data, you can execute the statements in the file and export data to a CSV file.

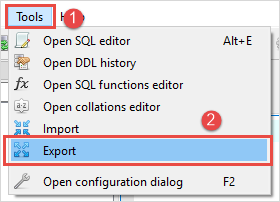
>sqlite3 -header -csv c:/sqlite/chinook.db < query.sql > data.csv

## Export SQLite database to a CSV file using SQliteStudio

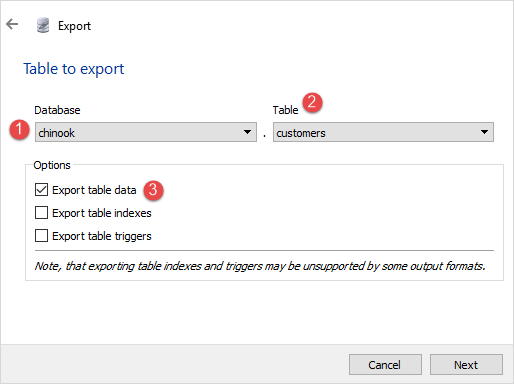
The SQLiteStudio provides the export function that allows you to export data in a table or the result of a query to a CSV file.

The following steps show you how to export data from a table to a CSV file.

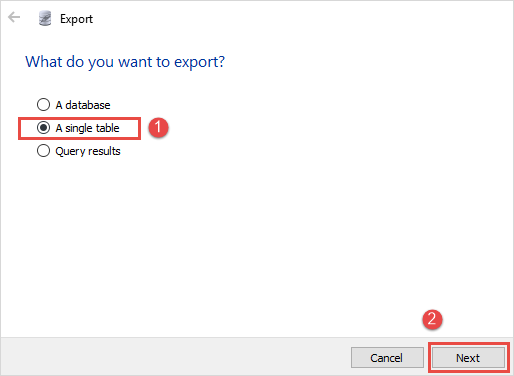
First, click the **Tools > Export** menu item



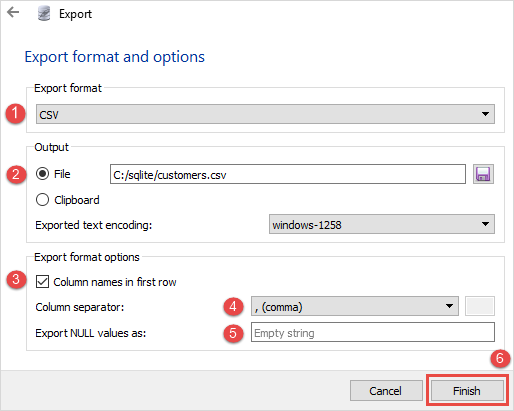
Next, choose the database and table that you want to export data; check the Export table data.



Then, choose a single table to export the data.



After that, (1) choose the CSV as the export format, (2) specify the CSV file name, (3) check the column names in the first row, (4) choose comma (,) as the column separator, (5) treat the NULL value as empty string, (6) click Finish button to complete exporting.



Finally, check the customer.csv file, you will see the following content:

