



# 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>
```

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.

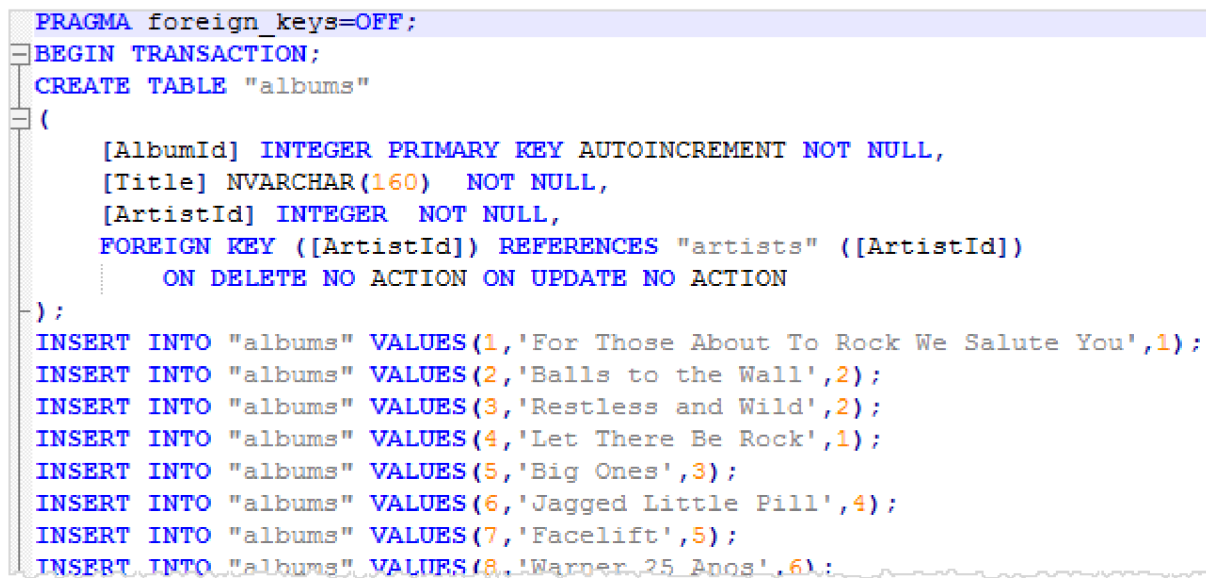
```
sqlite> .output c:/sqlite/chinook.sql
sqlite> .dump
sqlite> .exit
```

## 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
```

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



```
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE "albums"
(
  [AlbumId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
  [Title] NVARCHAR(160) NOT NULL,
  [ArtistId] INTEGER NOT NULL,
  FOREIGN KEY ([ArtistId]) REFERENCES "artists" ([ArtistId])
  ON DELETE NO ACTION ON UPDATE NO ACTION
);
INSERT INTO "albums" VALUES(1,'For Those About To Rock We Salute You',1);
INSERT INTO "albums" VALUES(2,'Balls to the Wall',2);
INSERT INTO "albums" VALUES(3,'Restless and Wild',2);
INSERT INTO "albums" VALUES(4,'Let There Be Rock',1);
INSERT INTO "albums" VALUES(5,'Big Ones',3);
INSERT INTO "albums" VALUES(6,'Jagged Little Pill',4);
INSERT INTO "albums" VALUES(7,'Facelift',5);
INSERT INTO "albums" VALUES(8,'Warner 25 Anos',6);
```

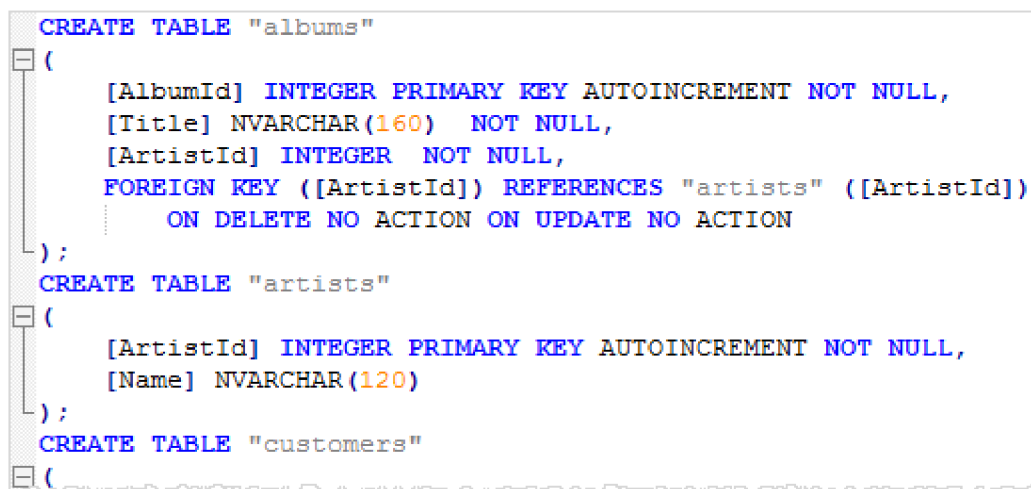
## Dump tables structure only using schema command

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
```

The following picture shows the content of the `chinook_structure.sql` file.



```
CREATE TABLE "albums"
(
    [AlbumId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    [Title] NVARCHAR(160) NOT NULL,
    [ArtistId] INTEGER NOT NULL,
    FOREIGN KEY ([ArtistId]) REFERENCES "artists" ([ArtistId])
        ON DELETE NO ACTION ON UPDATE NO ACTION
);
CREATE TABLE "artists"
(
    [ArtistId] INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL,
    [Name] NVARCHAR(120)
);
CREATE TABLE "customers"
(
```

## 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
```

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
```

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;
```

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

```
INSERT INTO table VALUES(1,'AC/DC');  
INSERT INTO table VALUES(2,'Accept');  
INSERT INTO table VALUES(3,'Aerosmith');  
INSERT INTO table VALUES(4,'Alanis Morissette');  
INSERT INTO table VALUES(5,'Alice In Chains');  
INSERT INTO table VALUES(6,'Antônio Carlos Jobim');  
INSERT INTO table VALUES(7,'Apocalyptica');  
INSERT INTO table VALUES(8,'Audioslave');  
INSERT INTO table VALUES(9,'BackBeat');  
INSERT INTO table VALUES(10,'Billy Cobham');
```

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

In this tutorial, you have learned how to dump data into a text file using the SQLite dump command and other commands.