

Unit-2: Database backup and CSV handling:**SQLite Dump:**

- The SQLite dump command to backup and restore a database.
- By using SQLite Dump command we can save the structure and data of the database tables in SQL format to the disk.

Syntax: sqlite> .dump

Example:**1) Dump the entire database into a file**

- 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.

Step-1 SQLite connection to the student.db file

```
C:\Users\ADMIN>sqlite3 student.db
SQLite version 3.42.0 2023-05-16 12:36:15
Enter ".help" for usage hints.
```

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

```
sqlite> .output c:/sqlite3/ student.sql
sqlite> .dump
sqlite> .exit
```

2) Dump specific table into file:

- If you want to dump a specific table, you need to specify the table name followed the .dump command.
- For example, the following command saves the product table to the product.sql file.

1. Dump only one table from the database:

Step-1: Open the database from which you want to dump a table. You can open a data base from following statement.

```
sqlite> Sqlilte3 db1.db
```

Here db1 is a name of the database.

Step-2: Write down following step to dump a specific table into file.

```
sqlite>.output c:/sqlite/product.sql
sqlite>.dump product
sqlite>.quit
```

Here, product.sql is name of the file in which you want to dump a table.

2. Dump table structure**Example:**

```
sqlite>.output c:/sqlite/student.sql
sqlite>.schema
```

3. Dump data of one or more tables into a file

```
sqlite>.mode insert  
sqlite>.output record.sql  
sqlite>select * from student;
```

CSV File Handling:

1. Import a CSV File Into an SQLite Table

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

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

CREATE TABLE cities("name"TEXT, "population"TEXT);

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

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

2. Export SQLite Database To a CSV File

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:
- Turn on the header of the result set using the .header on command.
- Set the output mode to CSV to instruct the sqlite3 tool to issue the result in the CSV mode.
- Send the output to a CSV file.
- 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/company.db
- sqlite> .headers on
- sqlite> .mode csv
- sqlite> .output data.csv
- sqlite> SELECT * FROM customers;
- sqlite> .quit