Database Forensics

Module 09

Database Forensics

Databases are the primary source of electronic evidence for every organization irrespective of its size and complexity of the database. Database forensics is the forensic study, relating to databases and its metadata. In database forensics, the principles of computer forensics can be applied.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Tools demonstrated in this lab are available in C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics

Lab Scenario

Amy, working as an incident handling manager with a software development company, was notified of their products being available in the market before the official release. As an incident manager, she suspected a security breach and reported this to the FBI.

An investigation team of cybercrime experts visited the firm and started their initial investigation. Later, the team found out that some unknown persons had hacked the database to steal the products, and they also suspected that someone from inside the company had helped the perpetrators.

Lab Objectives

The objective of this lab is to offer complete information on database forensics. The tasks include extracting information from different databases and performing forensics investigation on them.

Lab Environment

In this lab, you need:

- A computer running on **Windows 2012 virtual machine**
- A web browser with an **Internet** connection
- Administrative privileges to run tools

Lab Duration

Time: 50 Minutes

Overview of Database Forensics

Databases contain the evidences of various sizes required by the court of law to convict the culprit and prove them guilty. Investigators may use the timestamps to check and validate the user activities on the database. The investigator can also focus on identifying the transactions in a database system for fraud verification.

It is always recommended to outline and define policies and procedures to be followed to carry out analysis during database forensics. It is also recommended to retrieve and analyze the data without causing any damage, ensuring its authenticity.

TASK 1

Overview

Lab Tasks

Recommended labs to assist you in database forensics:

- Extracting the Databases of an Android Device using **Andriller**.
- Analyzing SQLite Databases using **DB Browser for SQLite**.
- Performing Forensic Investigation on a **MySQL Server Database**.

Lab Analysis

Analyze and document the results related to the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab

1

Extracting the Databases of an Android Device using Andriller

Andriller is an application that performs read-only, forensically sound, non-destructive acquisition from Android devices. Extractors and decoders produce reports in HTML and Excel (.xlsx) formats.

Lab Scenario

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Databases play a vital role in storing user and application information in an android device. The information stored in these databases includes Phonebook contacts, Call logs, SMS, Synchronized accounts, WhatsApp chat messages, Viber call logs (if installed on the device), Wi-Fi passwords, etc. During the process of forensic investigation, these databases can be acquired and examined in order to obtain crucial information related to the person who owns the acquired mobile device.

Being an expert forensic investigator, your main job is to acquire as much information as possible from the mobile device and analyze it in search of valuable information.

Lab Objectives

In this lab, you will learn how to extract databases and other sensitive information from an android emulator using **Andriller**.

Lab Environment

This lab requires:

- A **Windows Server 2012** virtual Machine.
- **Andriller** located in **C:\CHFI-Tools\CHFIv9 Module 13 Mobile Forensics\Mobile Forensics Tools\Andriller**.
- Administrative Privileges to run the tools.

Lab Duration

Time: 15 Minutes

Overview of the Lab

- Ensure that you are using an emulated Android device
- Extract the databases using **Andriller**

Lab Tasks

TASK 1

Install Andriller

1. Before beginning this lab, logon to **Windows Server 2012** virtual machine and create a folder named **Andriller** on desktop.
2. Launch **AVD Manager** from the **Apps** screen.
3. Select **Test_Emulator** and click **Start...**.

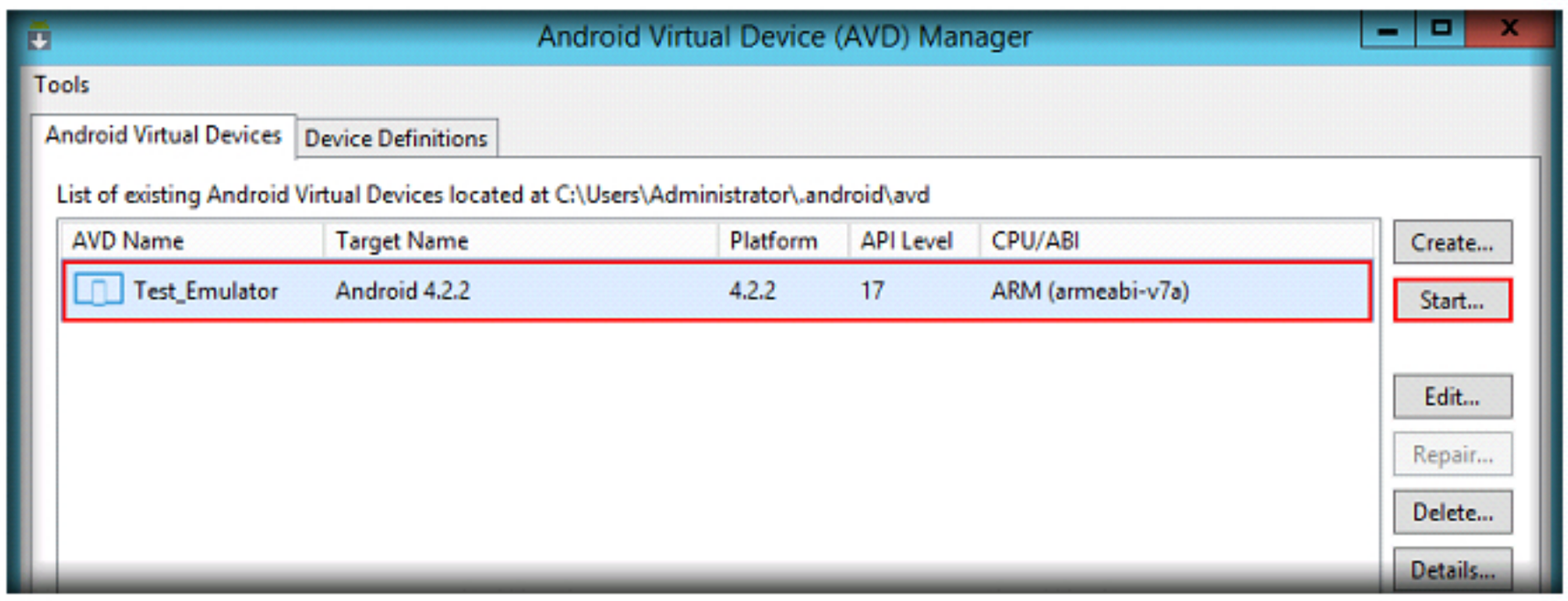

FIGURE 1.1: AVD Manager main window

4. **Launch Options** window appears. Check **Scale display to real size** option, specify the screen size as **6.5** inches and click **Launch**.

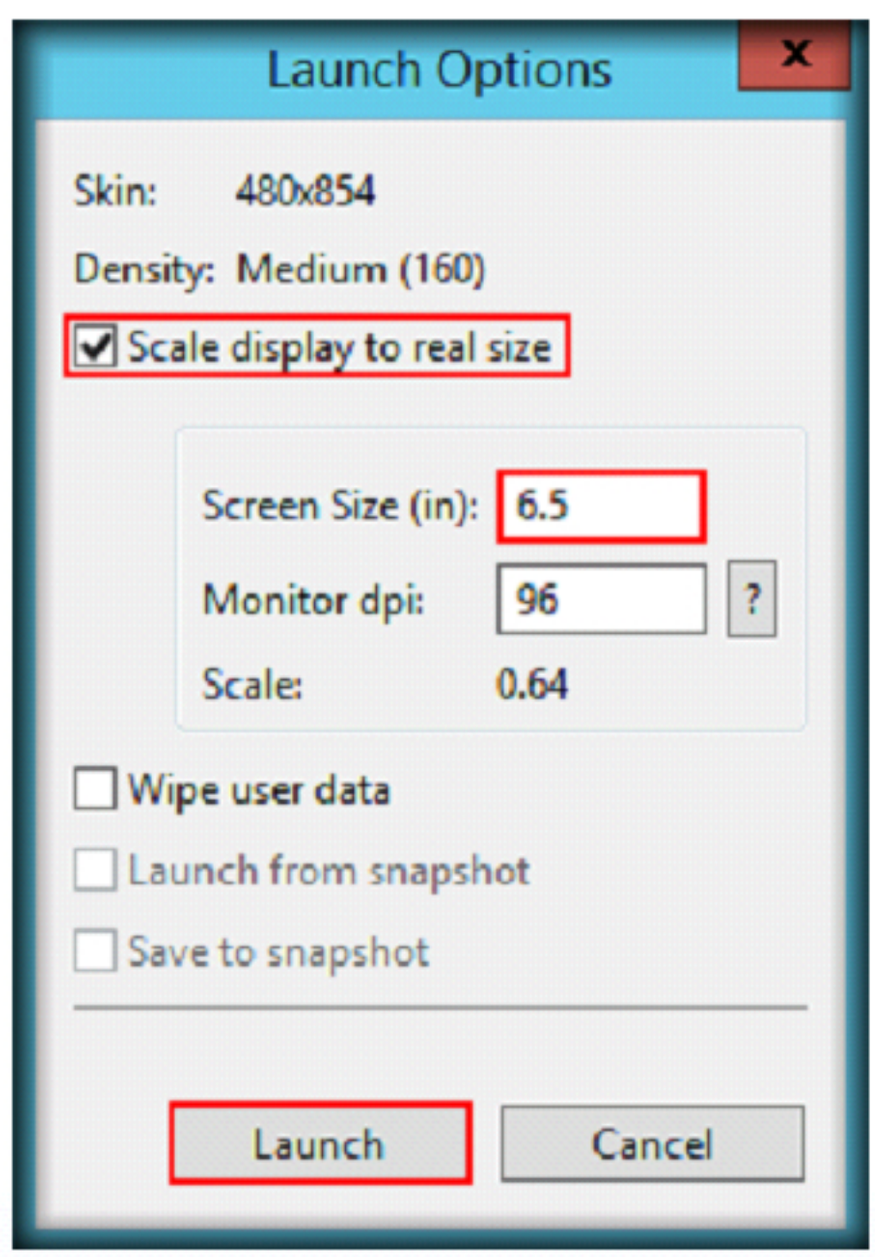

FIGURE 1.2: Launch Options window

5. Navigate to **C:\CHFI-Tools\CHFIv9 Module 13 Mobile Forensics\Mobile Forensics Tools\Andriller**, double-click **Andriller_v2.6.0.1_Setup.exe** and follow the wizard-driven installation steps to install the application.

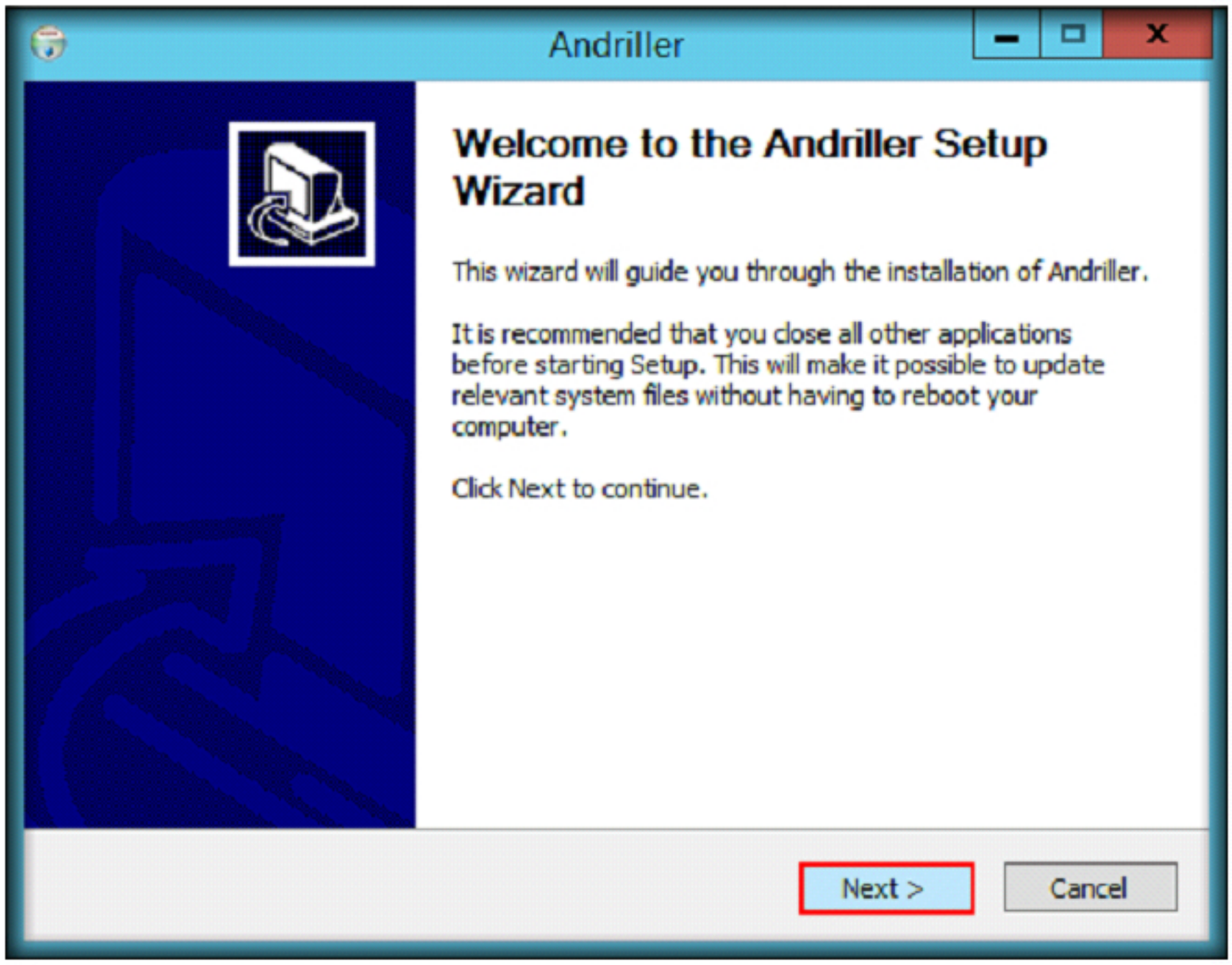

FIGURE 1.3: Installing Andriller

6. On completing the installation, launch Andriller application from the **Apps** screen.

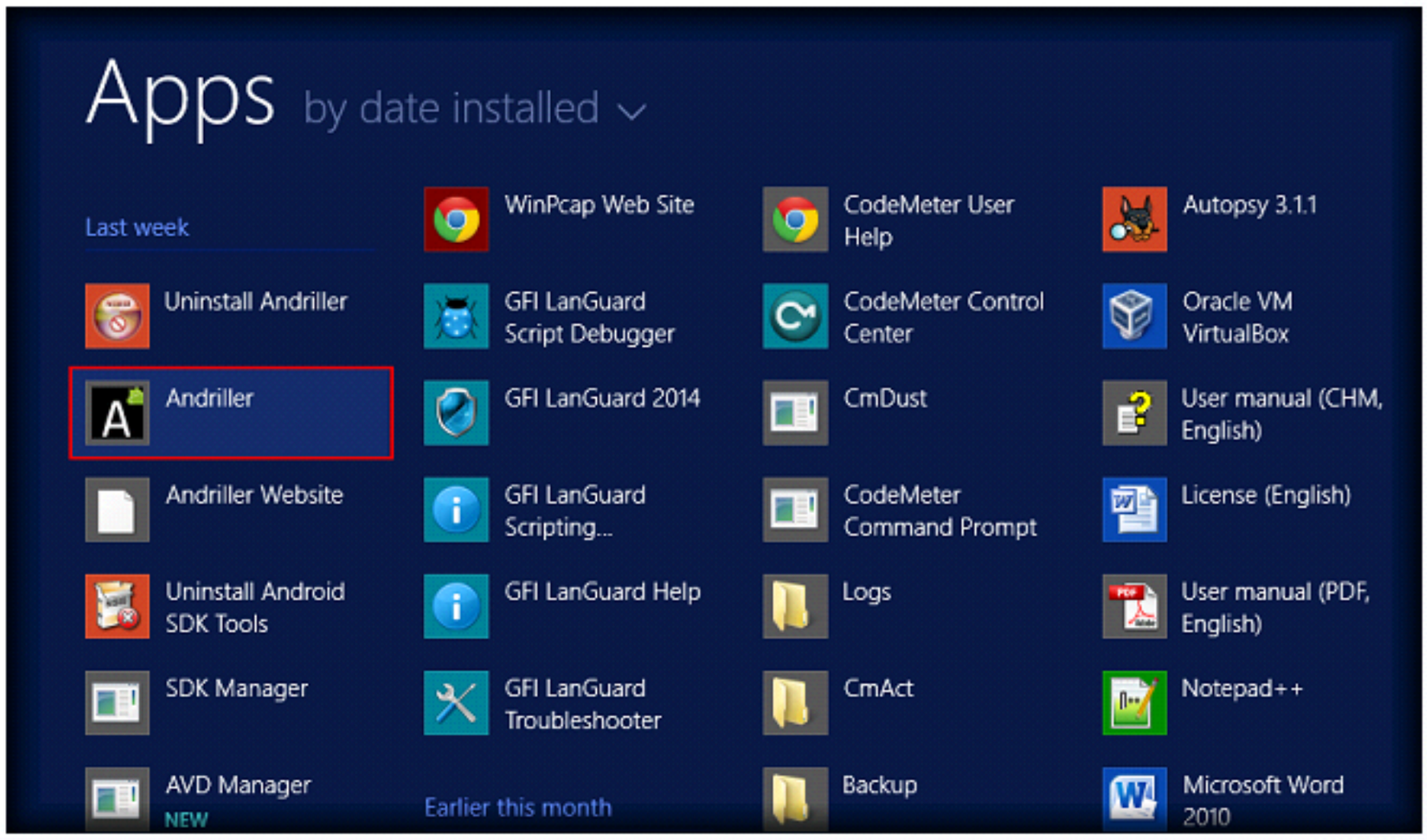

FIGURE 1.4: Launching Andriller

7. If a **Check New Versions?** dialog-box appears, click **No**.

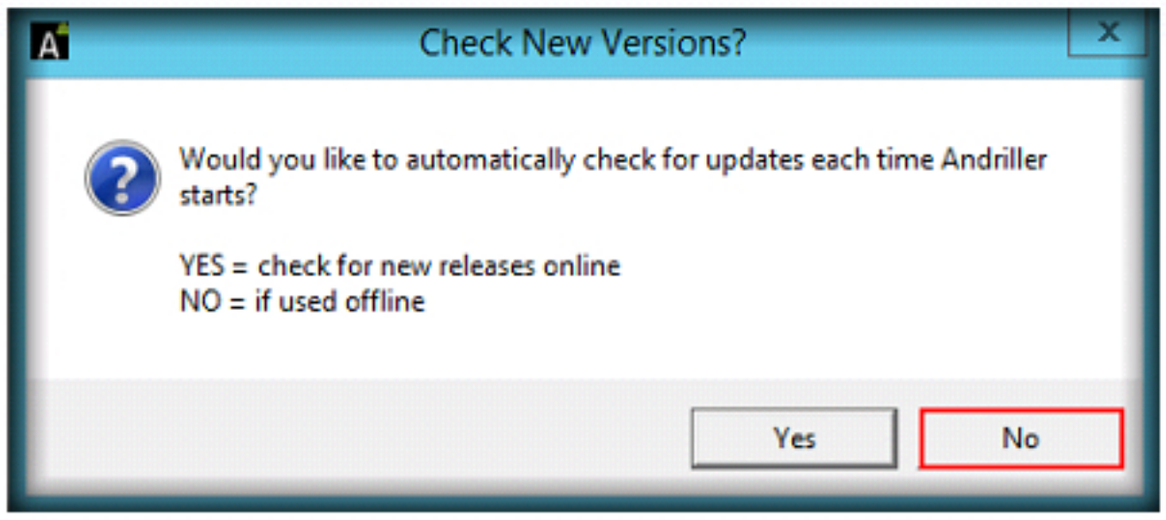

FIGURE 1.5: Check New Versions? dialog-box

8. If a **Preferences Menu** dialog-box appears, click **OK**.

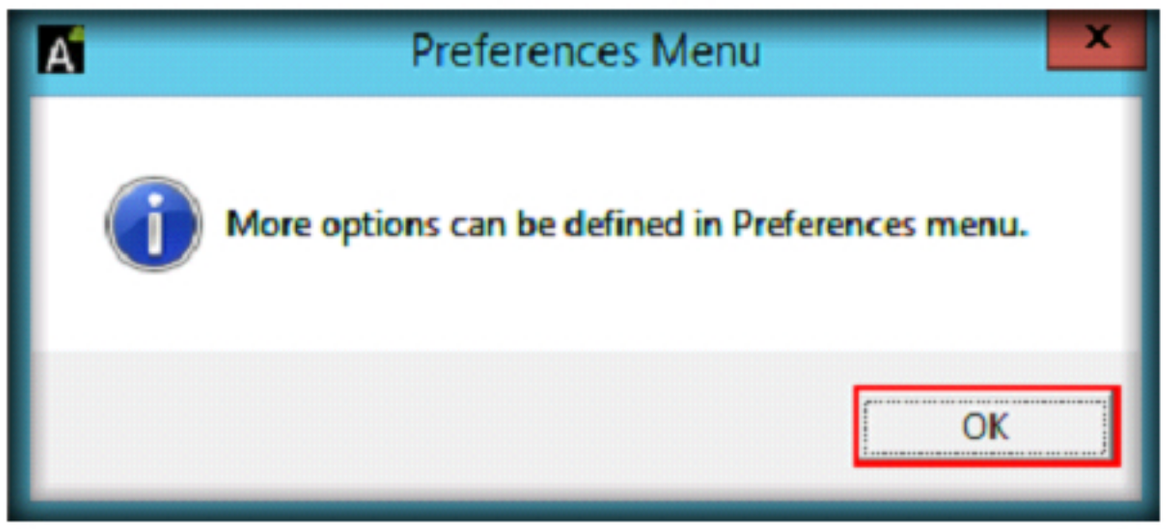

FIGURE 1.6: Preferences Menu dialog-box

9. Main window of Andriller appears as shown in the following screenshot:

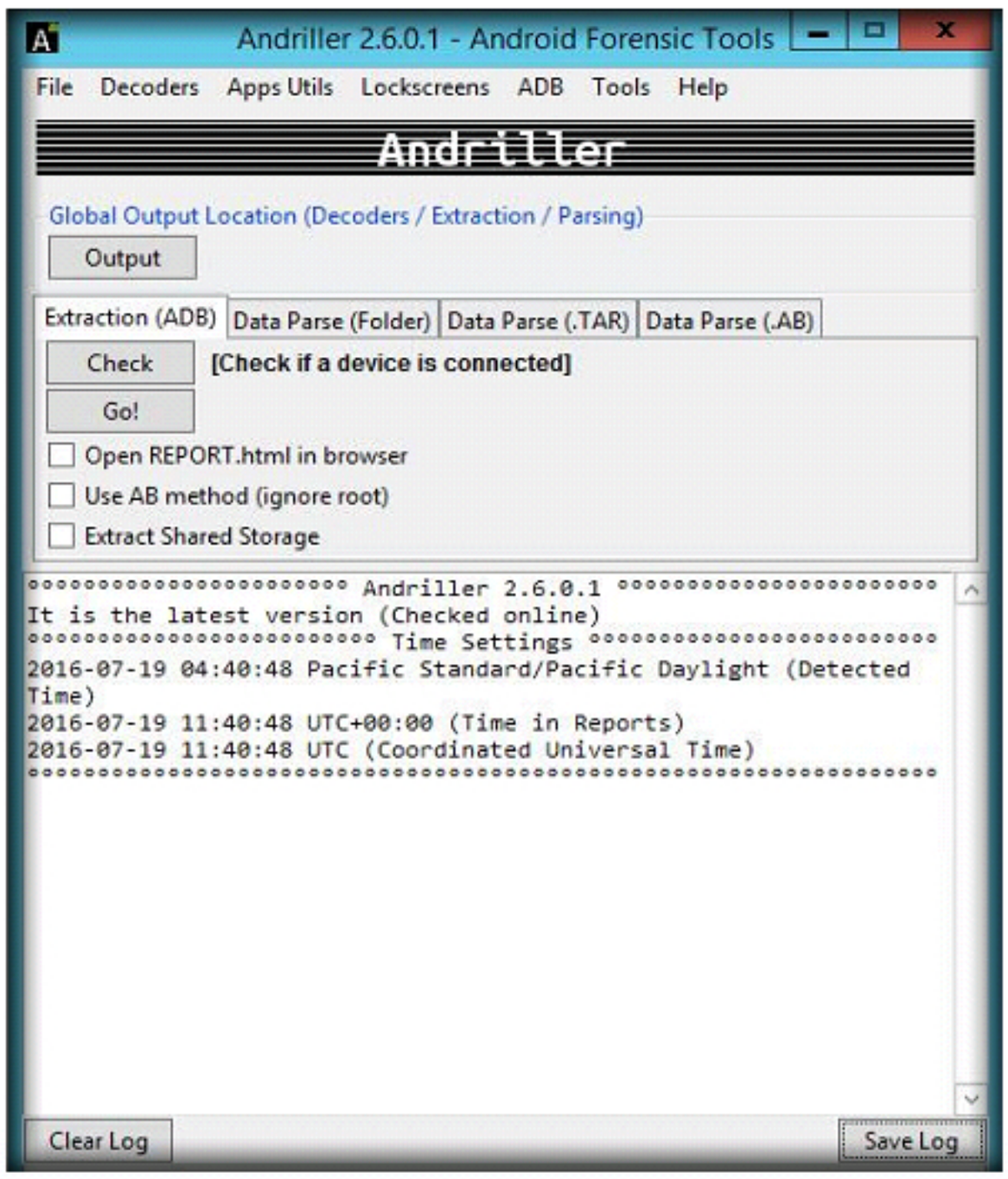

FIGURE 1.7: Andriller main window

TASK 2

Configure Andriller

10. You need to specify an output location for Andriller, where all the logs and data will be stored. Click **Output** button in the Andriller window.

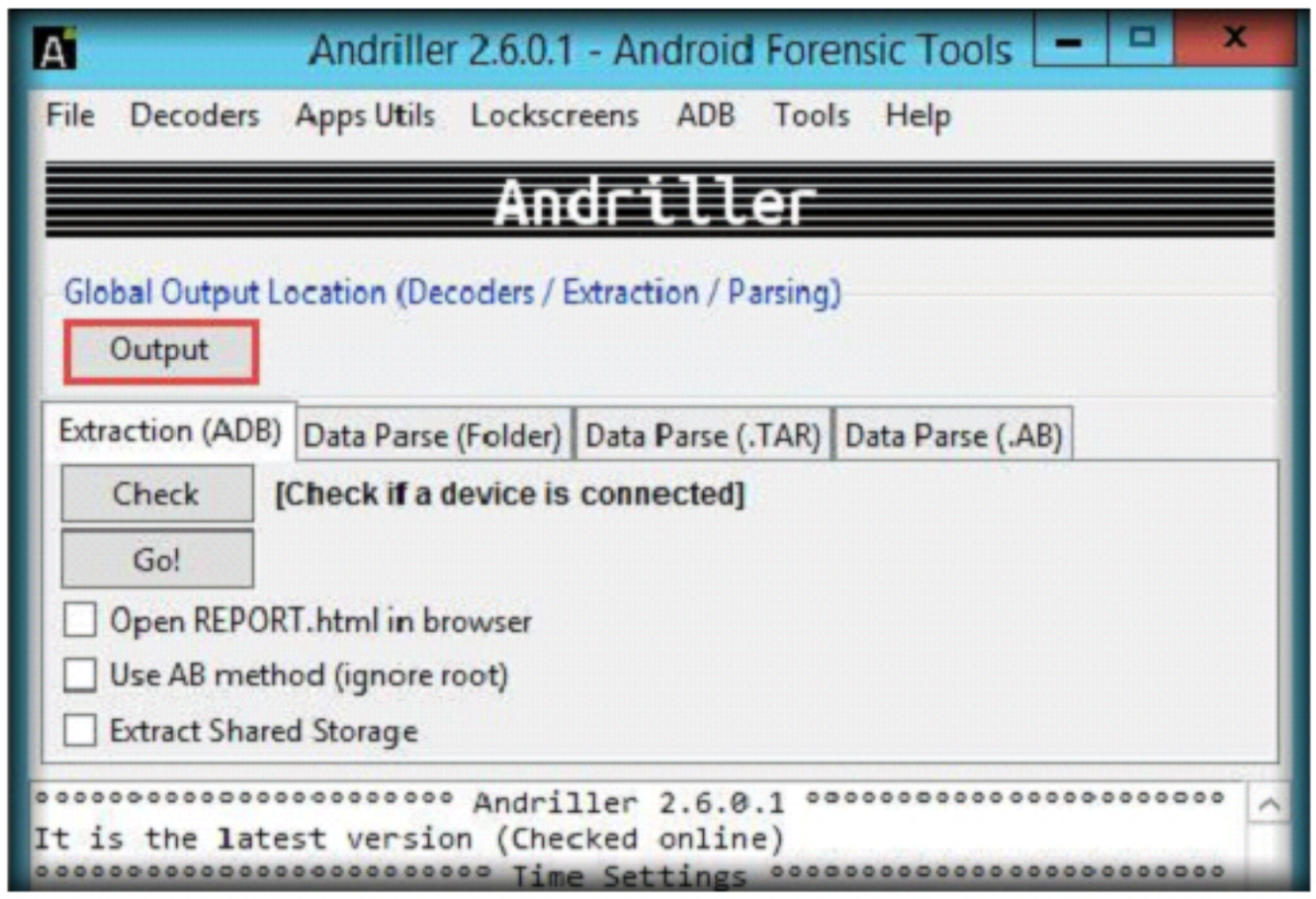

FIGURE 1.8: Configuring Output Folder

11. Navigate to **Desktop**, Select the **Andriller** folder that you created before installing the application and click **OK**.

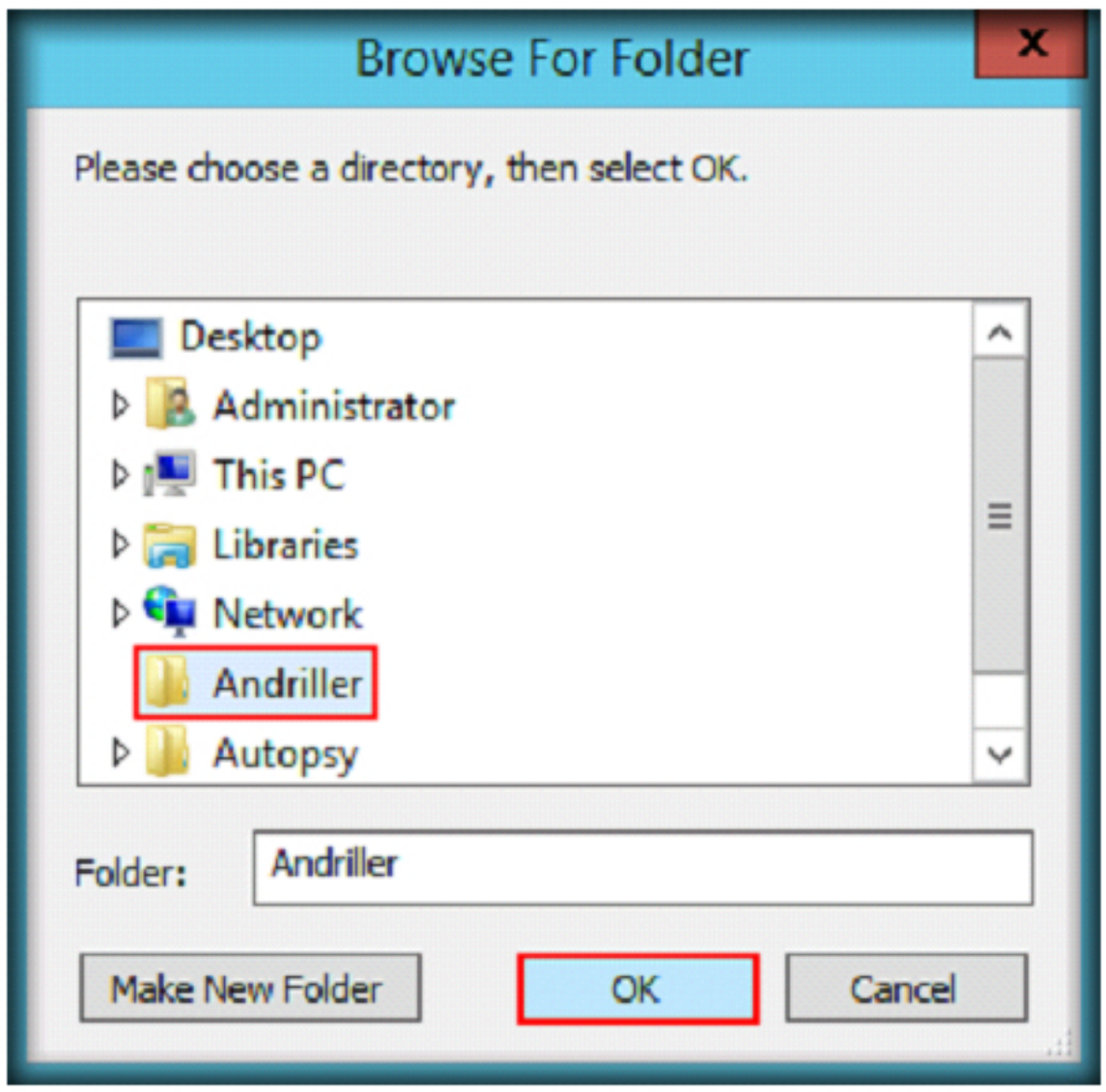

FIGURE 1.9: Configuring Output Folder

12. Enable **USB Debugging mode** in the rooted android device and **connect** it to the machine.

13. After connecting the device to the machine, click **Check** button in Andriller window.

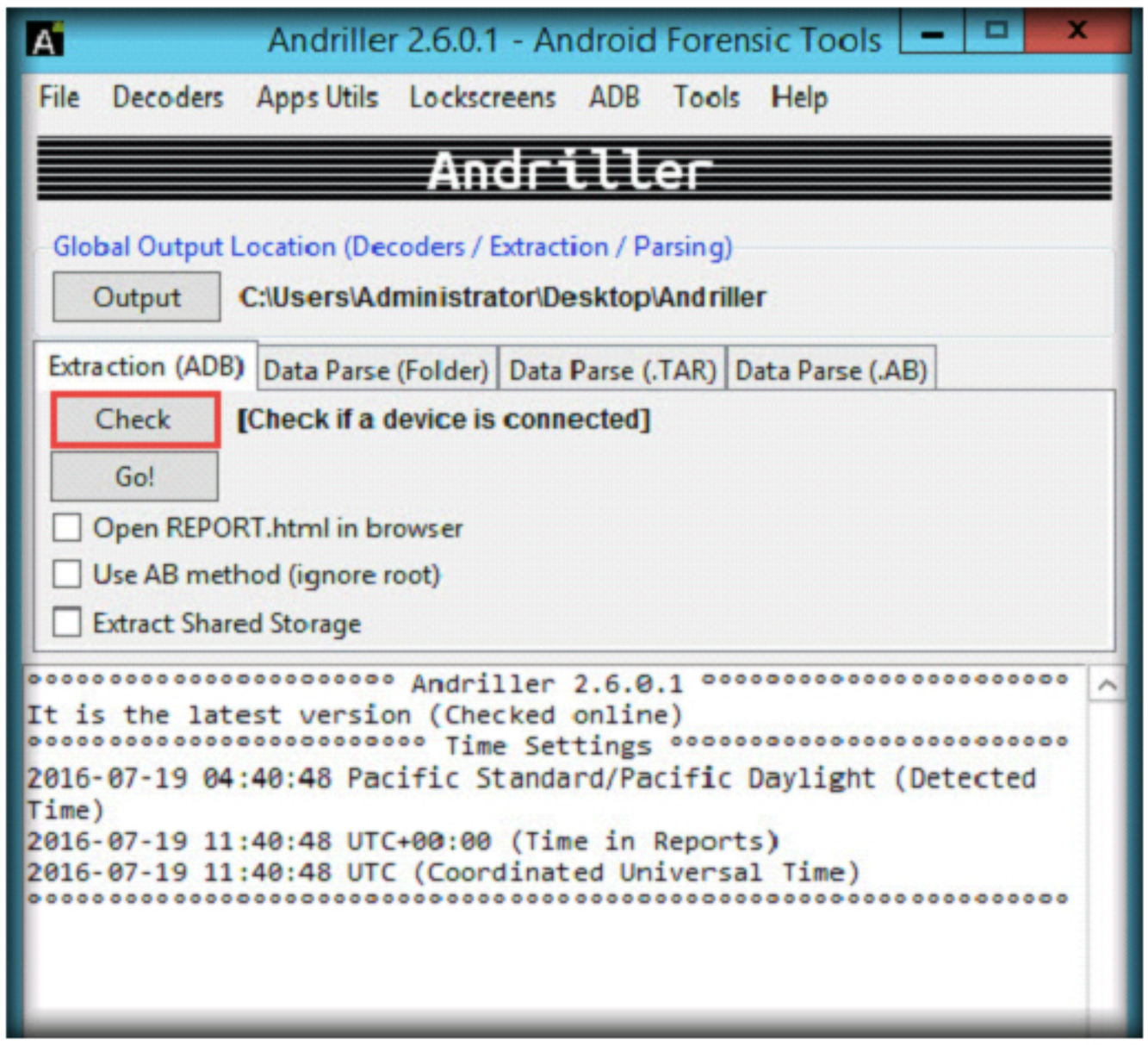

FIGURE 1.10: Testing for the Device

14. On clicking the **Check** button, Andriller should display a **serial ID** of the mobile device as shown in the following screenshot:

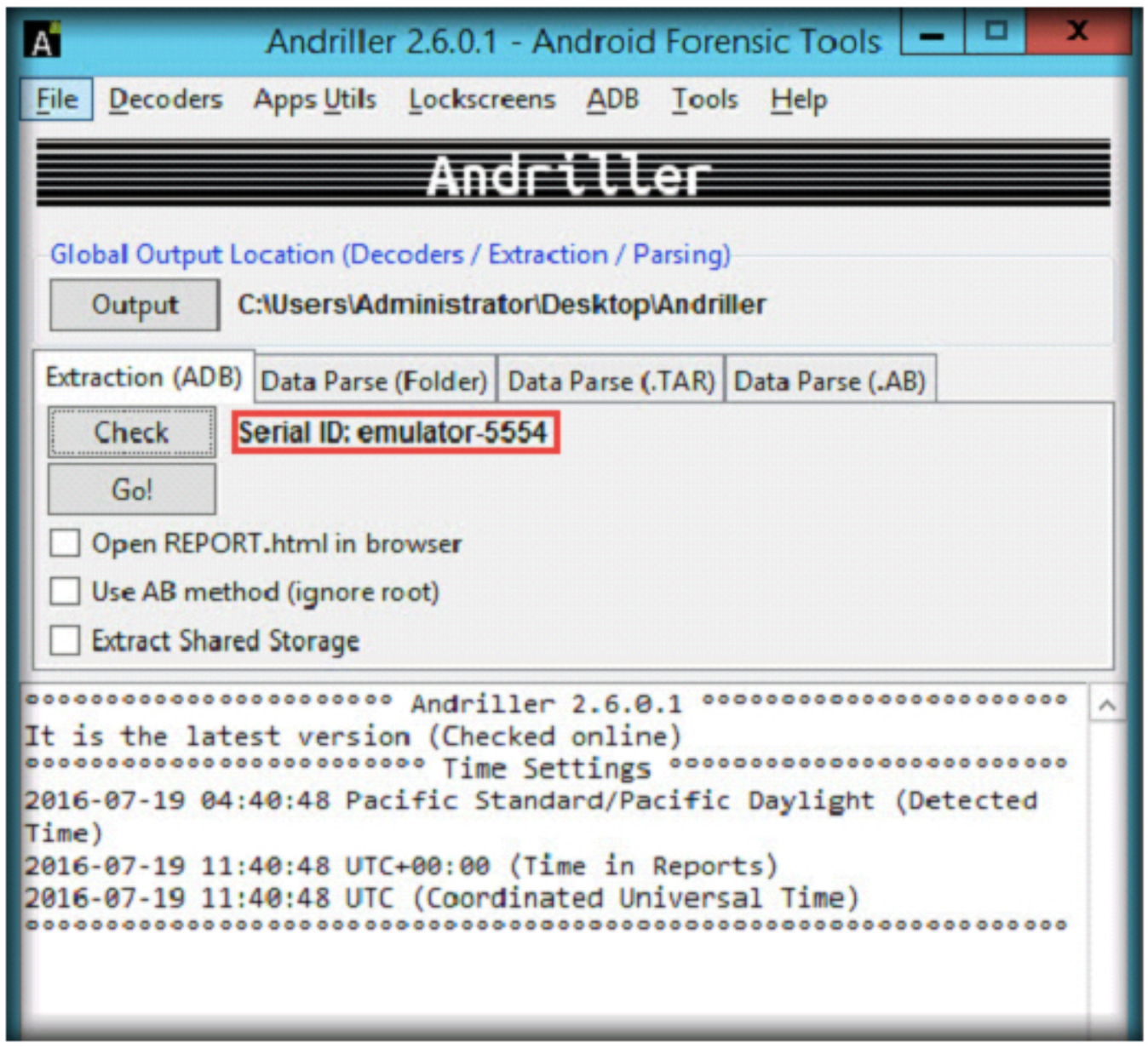

FIGURE 1.11: Device Successfully Detected

TASK 3

Begin Data Acquisition

15. Once the device is detected by Andriller, click **Go!** button to begin data extraction.

FIGURE 1.12: Beginning the Extraction

16. Andriller begins to extract the databases and other useful information as shown in the following screenshot:

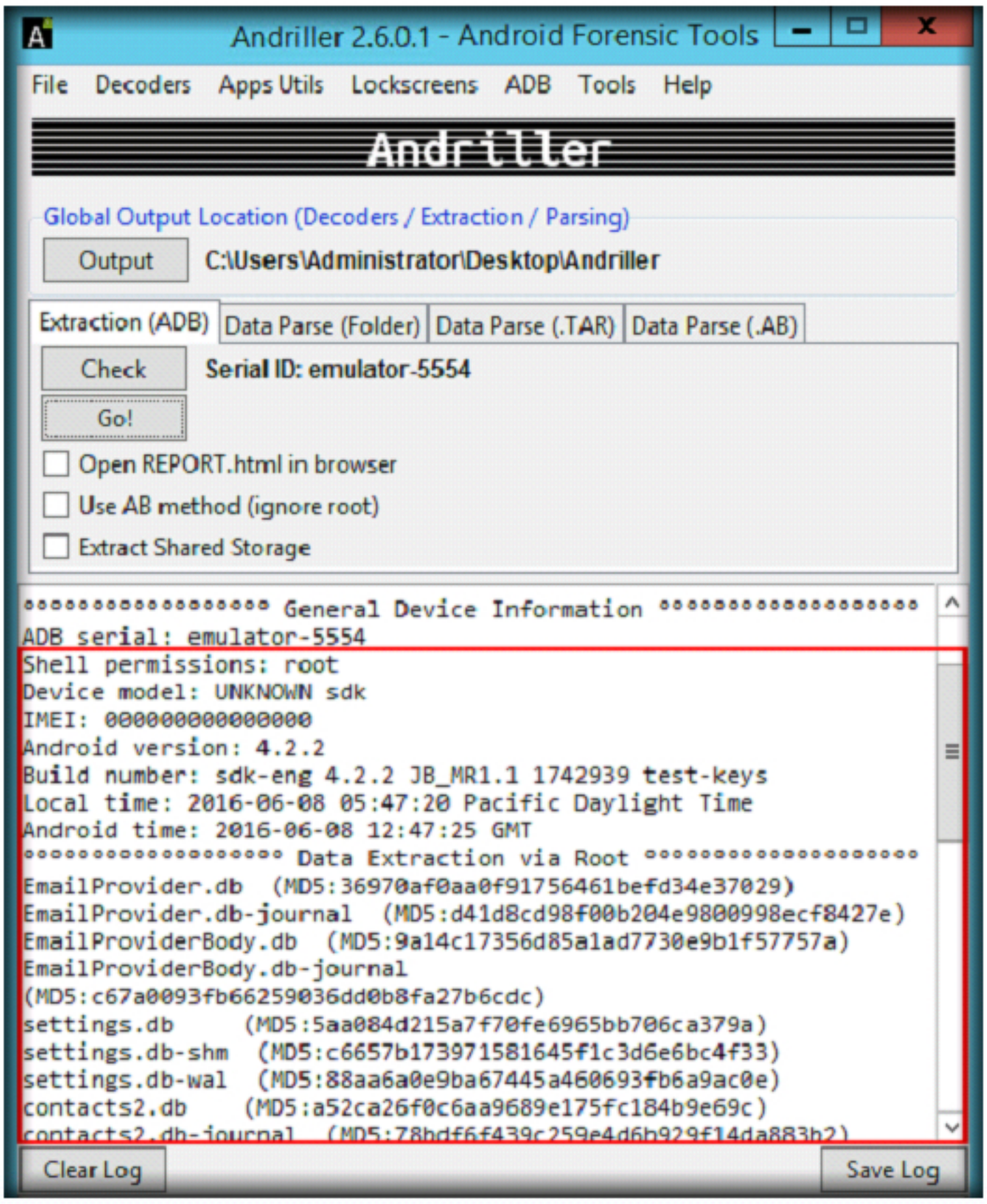

FIGURE 1.13: Extraction Initiated

17. Andriller creates a directory inside the **Andriller** folder with the name of the device followed by the timestamp as shown in the following screenshot:

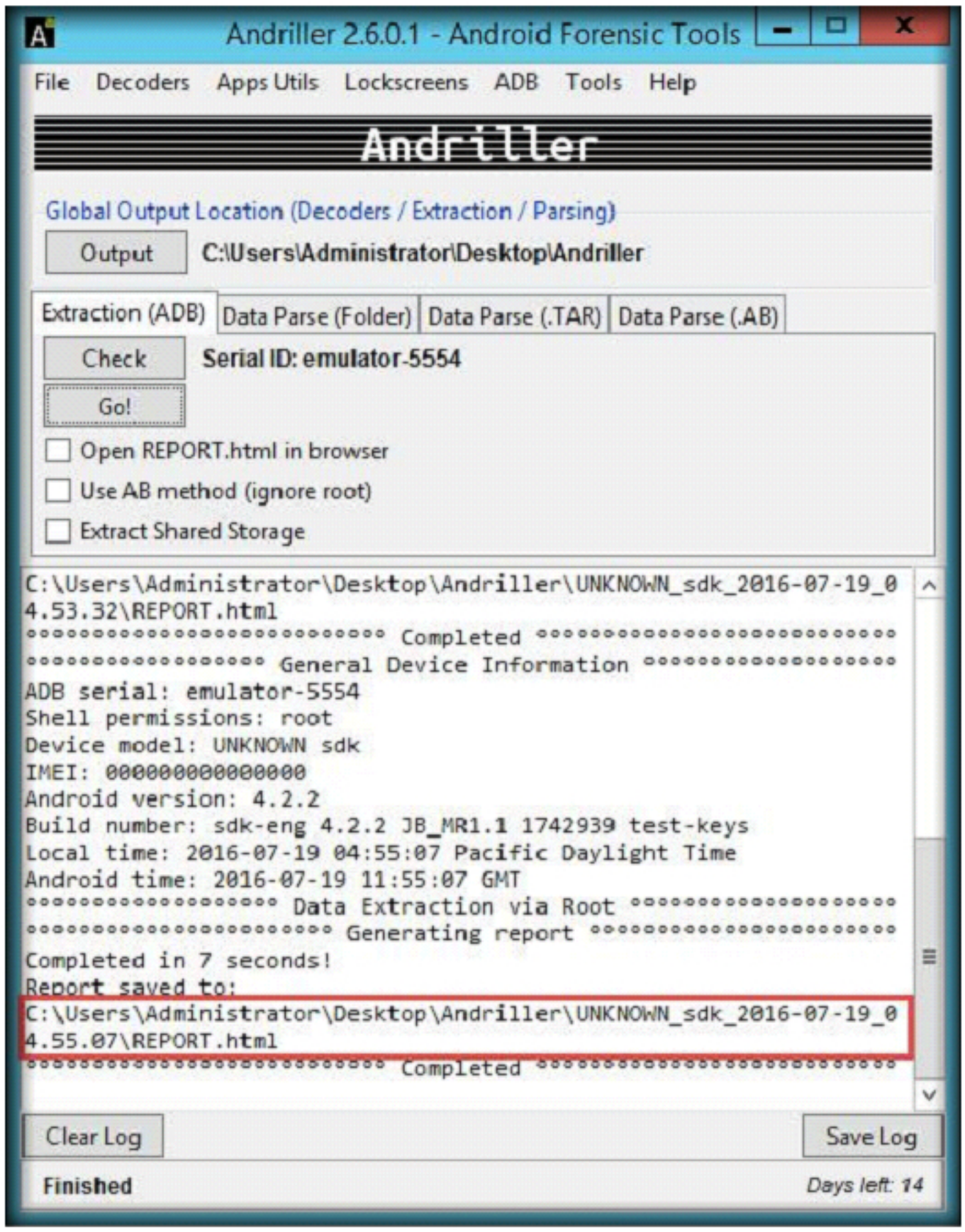

FIGURE 1.14: Extraction Completed

TASK 4

View the Acquired Files

Note: The folder name varies according to the device used in this lab.

18. Navigate to the **Andriller** folder located on **Desktop** and open the folder in which the extracted files and databases are stored.

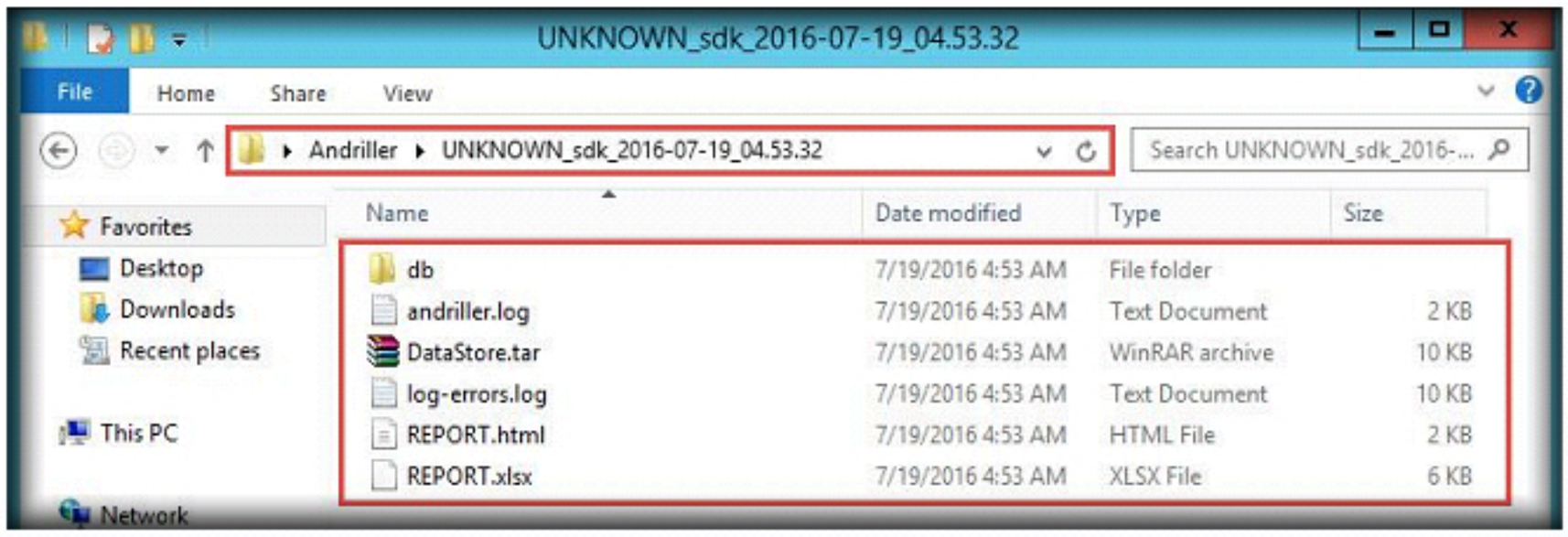

FIGURE 1.15: Viewing the Extracted Files

Lab Analysis

Analyze the result and document the findings of the lab.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab

2

Analyzing SQLite Databases using DB Browser for SQLite

DB Browser for SQLite is an open-source tool to create, design, and edit database files compatible with SQLite. It is for users and developers wanting to create databases, search, and edit data. It uses a familiar spreadsheet-like interface, and you don't need to learn complicated SQL commands.

ICON KEY

- Valuable information
- Test your knowledge
- Web exercise
- Workbook review

Lab Scenario

Ryan has lodged a complaint with the authorities regarding security breach in his gaming company that has caused him financial losses. The probe results included that the company was using SQLite Databases to store information and had failed to update the software. The investigators used a tool to find that the attacker had used a vulnerability scanner to hack into it.

As a forensic investigator, you should be aware of all the database technologies being used including tools required to analyze data in them. You can analyze the SQLite Databases using DB Browser for SQLite.

Lab Objectives

In this lab, you will learn how to analyze the SQLite Databases using the open-source tool **DB Browser for SQLite**.

Lab Environment

This lab requires:

- A **Windows Server 2012** virtual Machine.
- **DB Browser for SQLite** located in **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\DB Browser for SQLite**.
- An android device's databases.
- Administrative Privileges to run the tool.

Lab Duration

Time: 15 Minutes

Overview of the Lab

- Install and launch **DB Browser for SQLite**
- Examine the database files

Lab Tasks

TASK 1

Install Autopsy

1. In this lab, we will examine the databases extracted from an android device located at **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Databases for Analysis\SQLite Databases**.
2. Logon to **Windows Server 2012** virtual machine.
3. Navigate to **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\DB Browser for SQLite**, double-click **sqlitebrowser-3.8.0-win64v2.exe,** and follow the wizard-driven installation steps to install the application.

Note: If an **Open File Security Warning** pop-up appears, click **Run**.

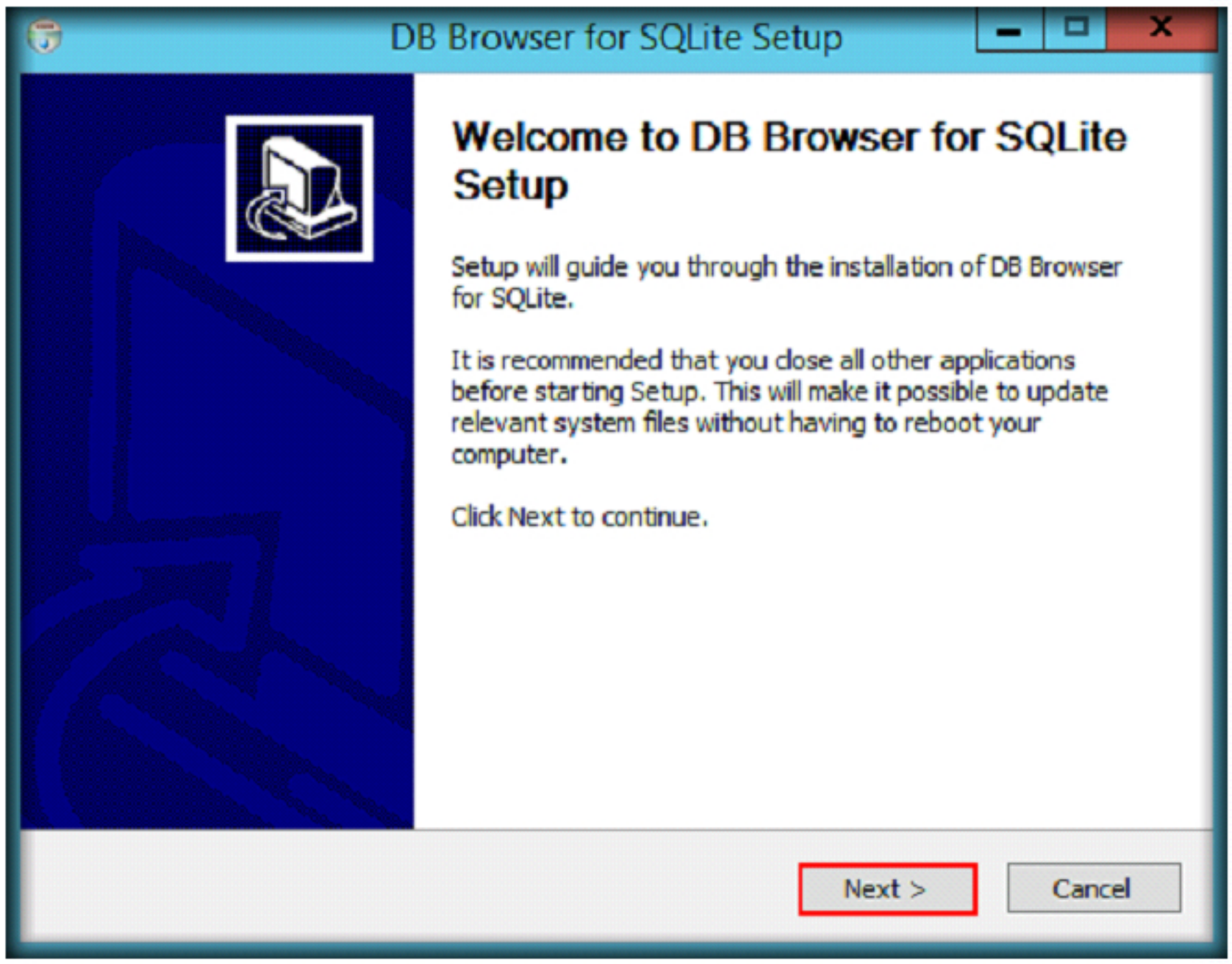

FIGURE 2.1: DB Browser for SQLite Setup Installation Wizard

4. In the final step of installation, ensure that **Run DB Browser for SQLite** option is checked and click **Finish**.

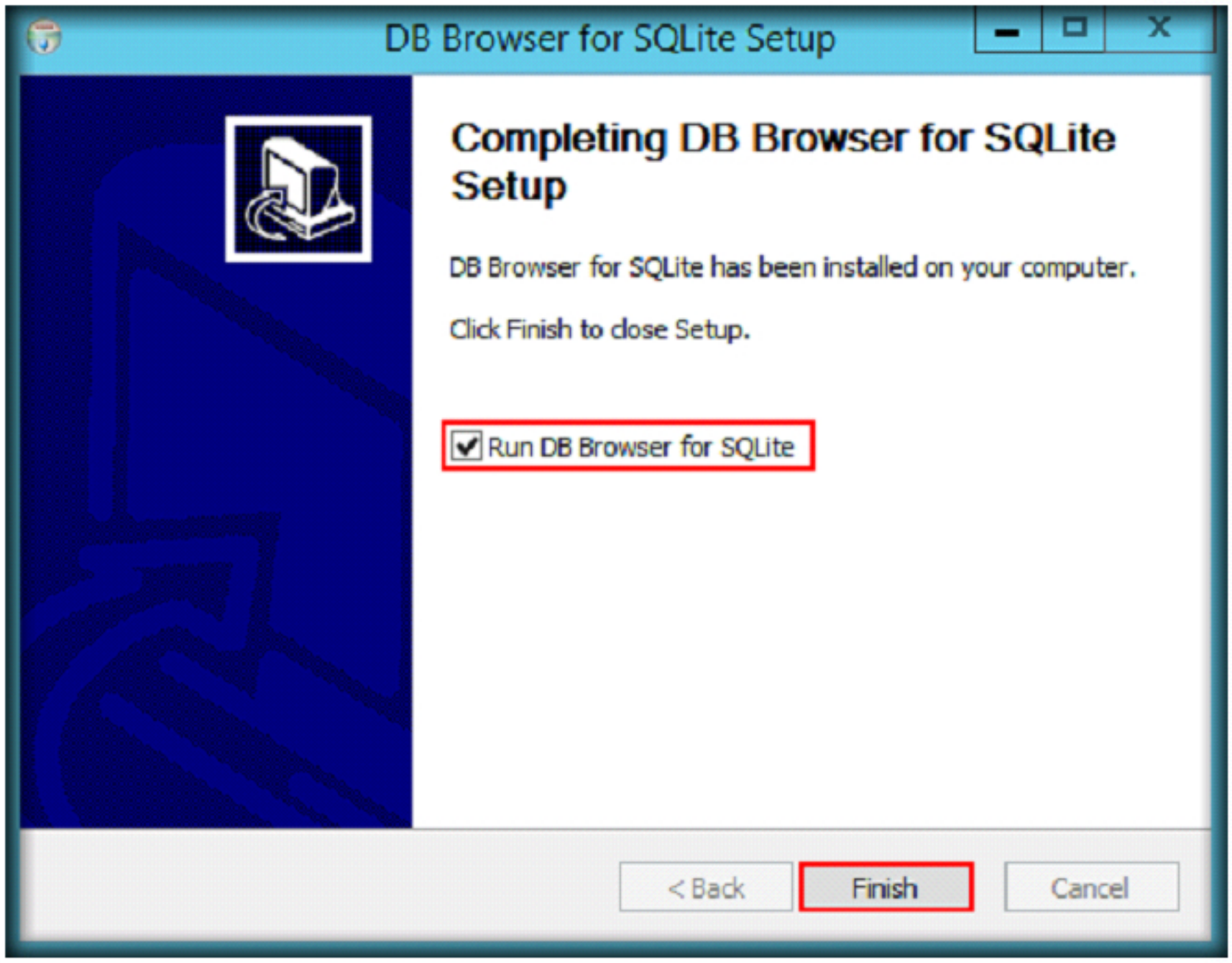

FIGURE 2.2: DB Browser for SQLite final step of installation

TASK 2

Create an Image

5. **DB Browser for SQLite** GUI appears. Click **Open Database** in the toolbar.

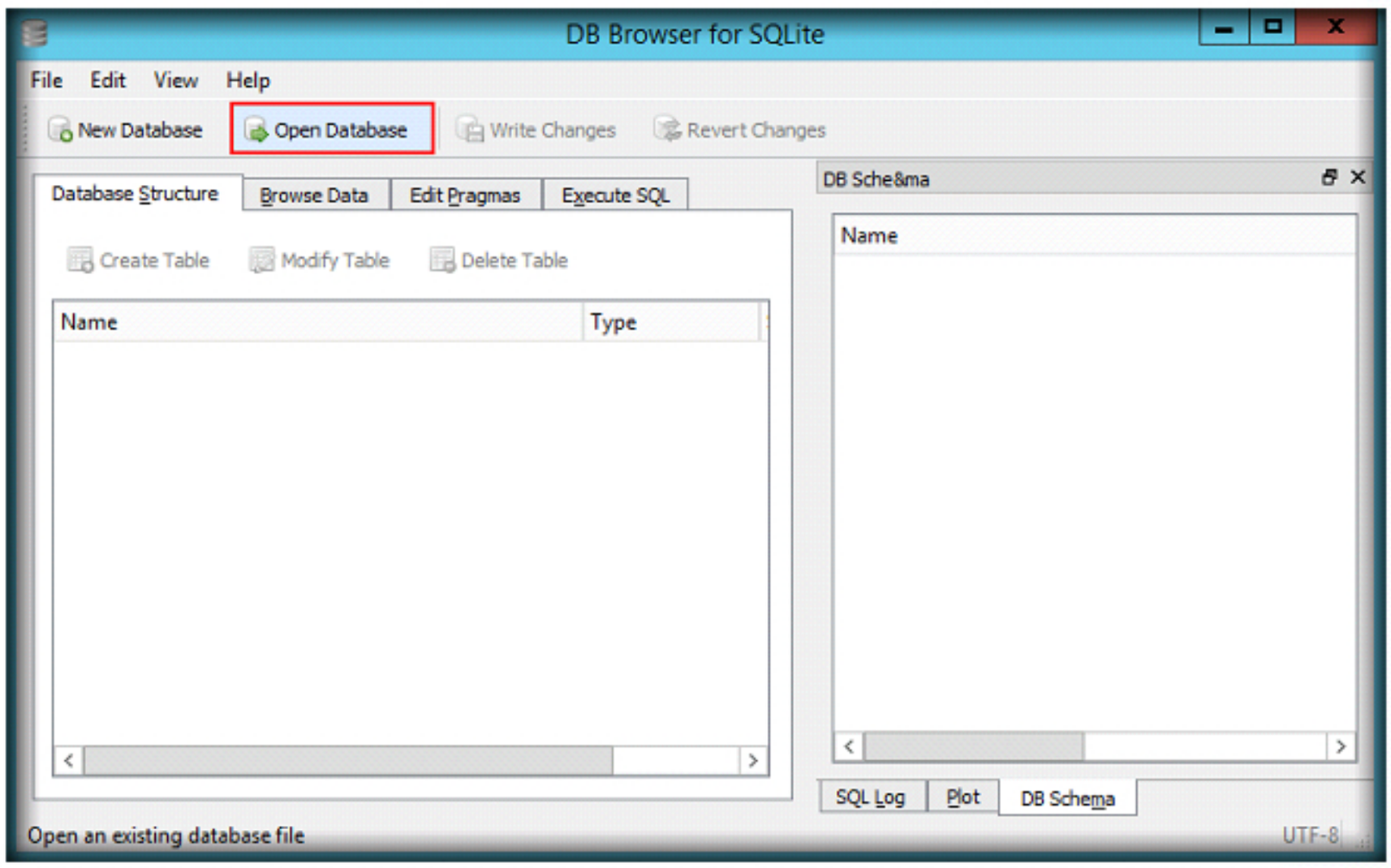

FIGURE 2.3: DB Browser for SQLite main window

6. **Choose a database file** window appears. Navigate to **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Databases for Analysis\SQLite Databases**, select **accounts.db**, and click **Open**.

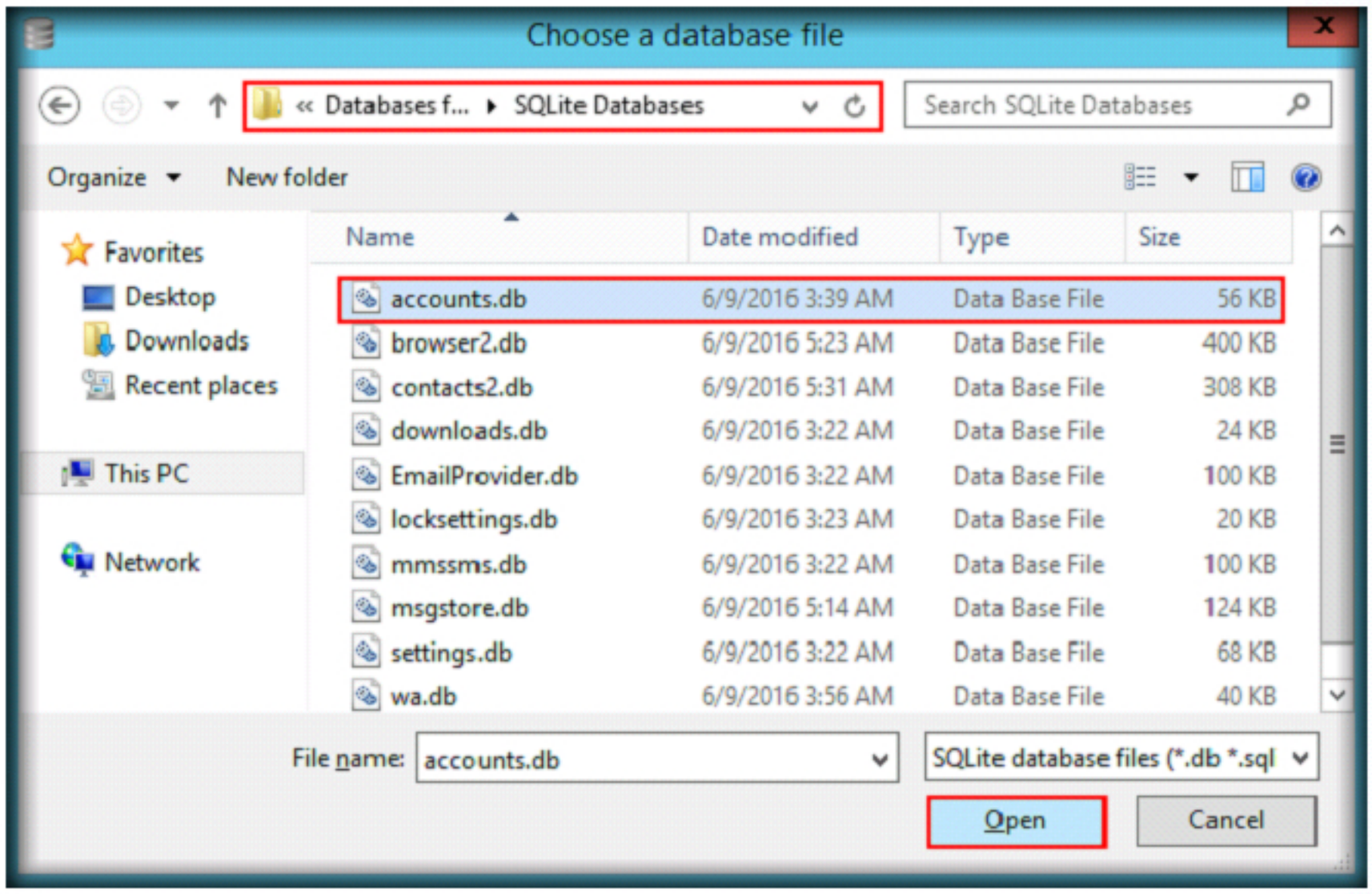

FIGURE 2.4: Choose a database file window

7. The application displays the structure of accounts database under the **Database Structure** tab as shown in the following screenshot:

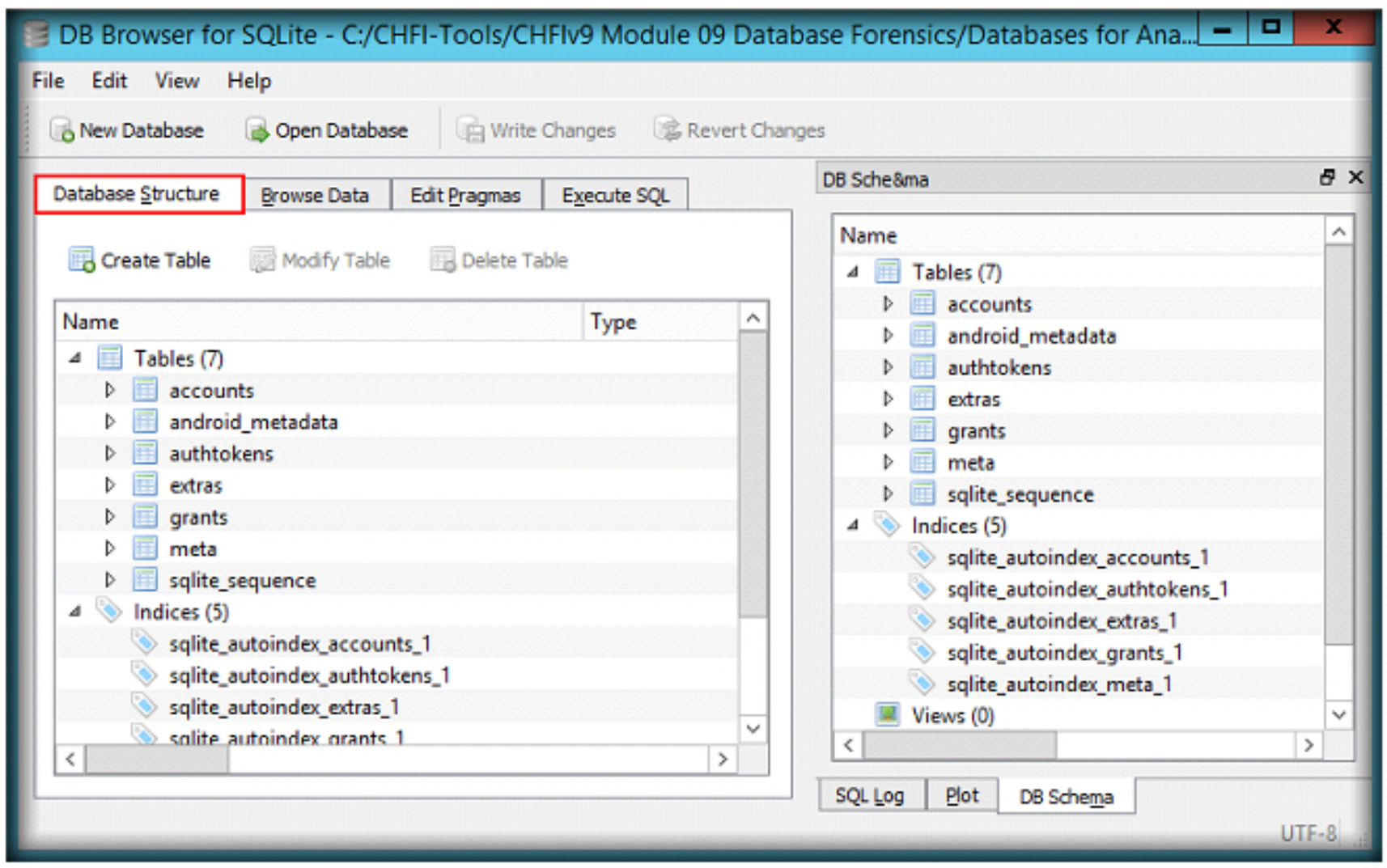

FIGURE 2.5: Screenshot showing Database Structure tab

8. Click **Browse Data** tab to view the data in the accounts database.
9. Once you click the tab button, the **accounts** table will be selected by default and the table contents (the accounts synchronized with the device) will be displayed under the **Table** section and the database schema will be displayed in the **right pane** of the UI as shown in the following screenshot:

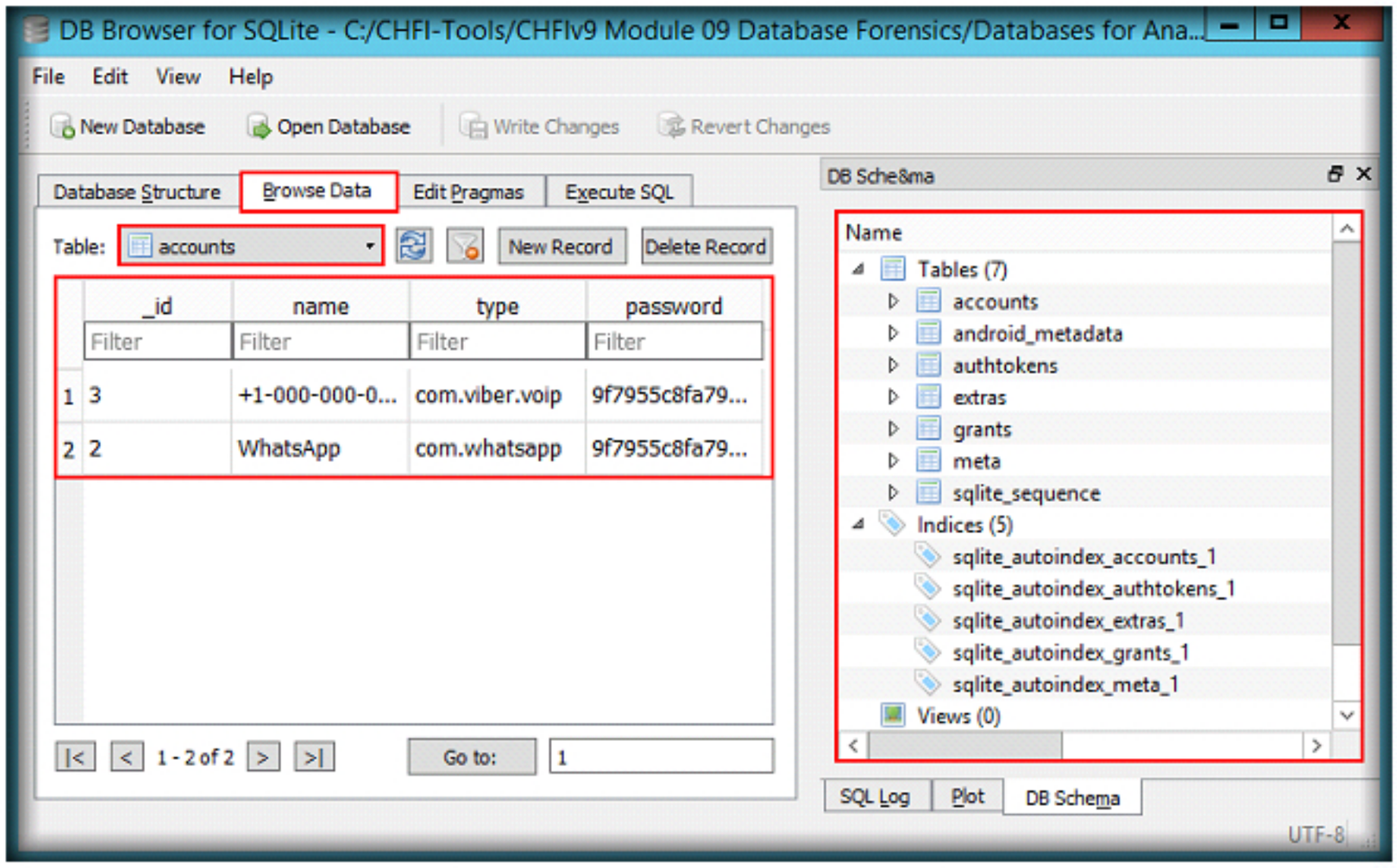

FIGURE 2.6: Screenshot showing Browse Data tab

10. We can observe that the device was synchronized with two accounts: WhatsApp and Viber.
11. In the same way, you may also view the contents of other tables by selecting them from the **Table** drop-down list.
12. Now, we shall view the information stored in the **browser** database. To go to the database, click **Open Database** from the toolbar.

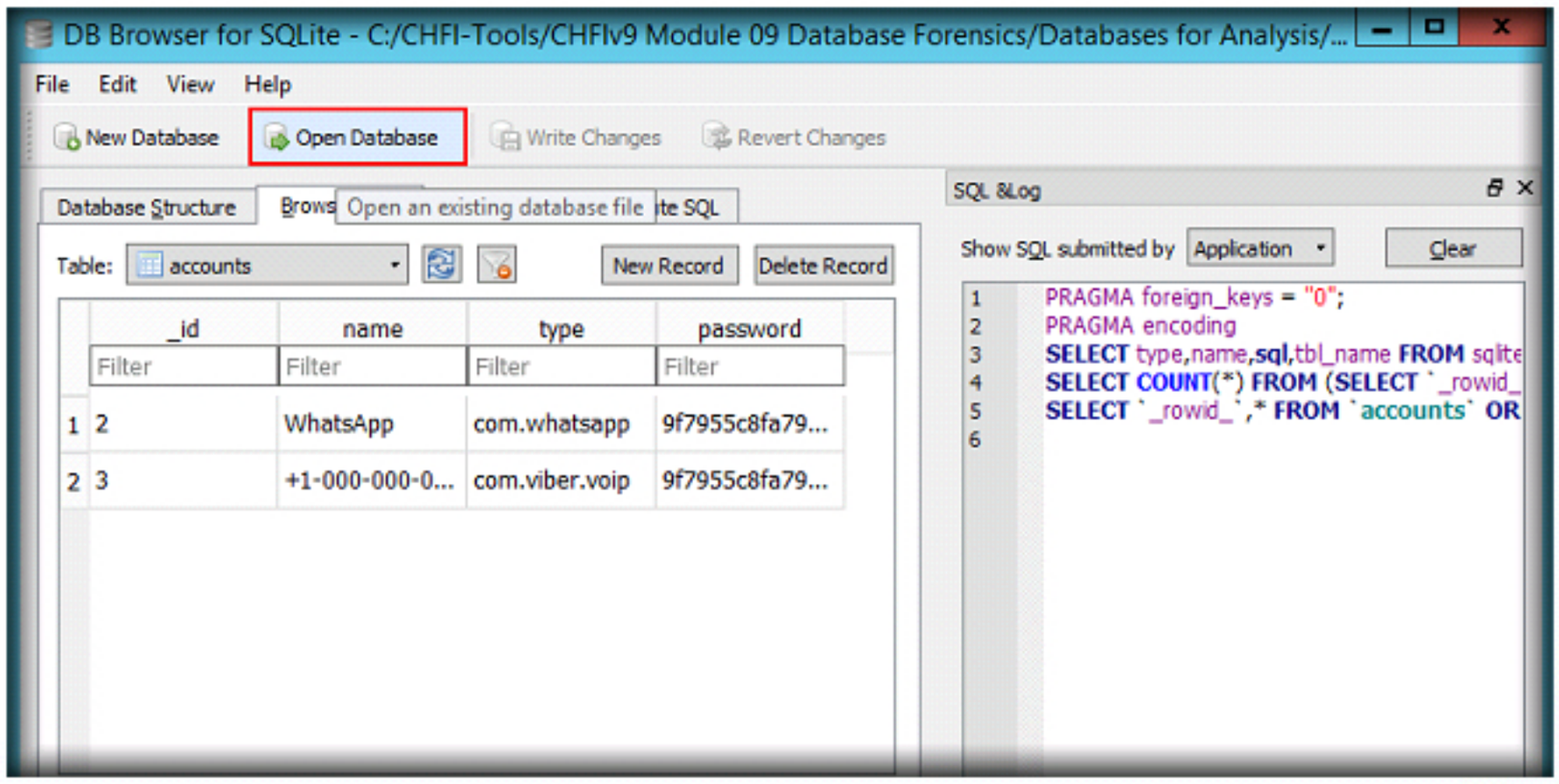

FIGURE 2.7: Screenshot showing Open Database option in the toolbar

13. **Choose a database file** window appears. Point to the location **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Databases for Analysis\SQLite Databases**, select **browser2.db**, and click **Open**.

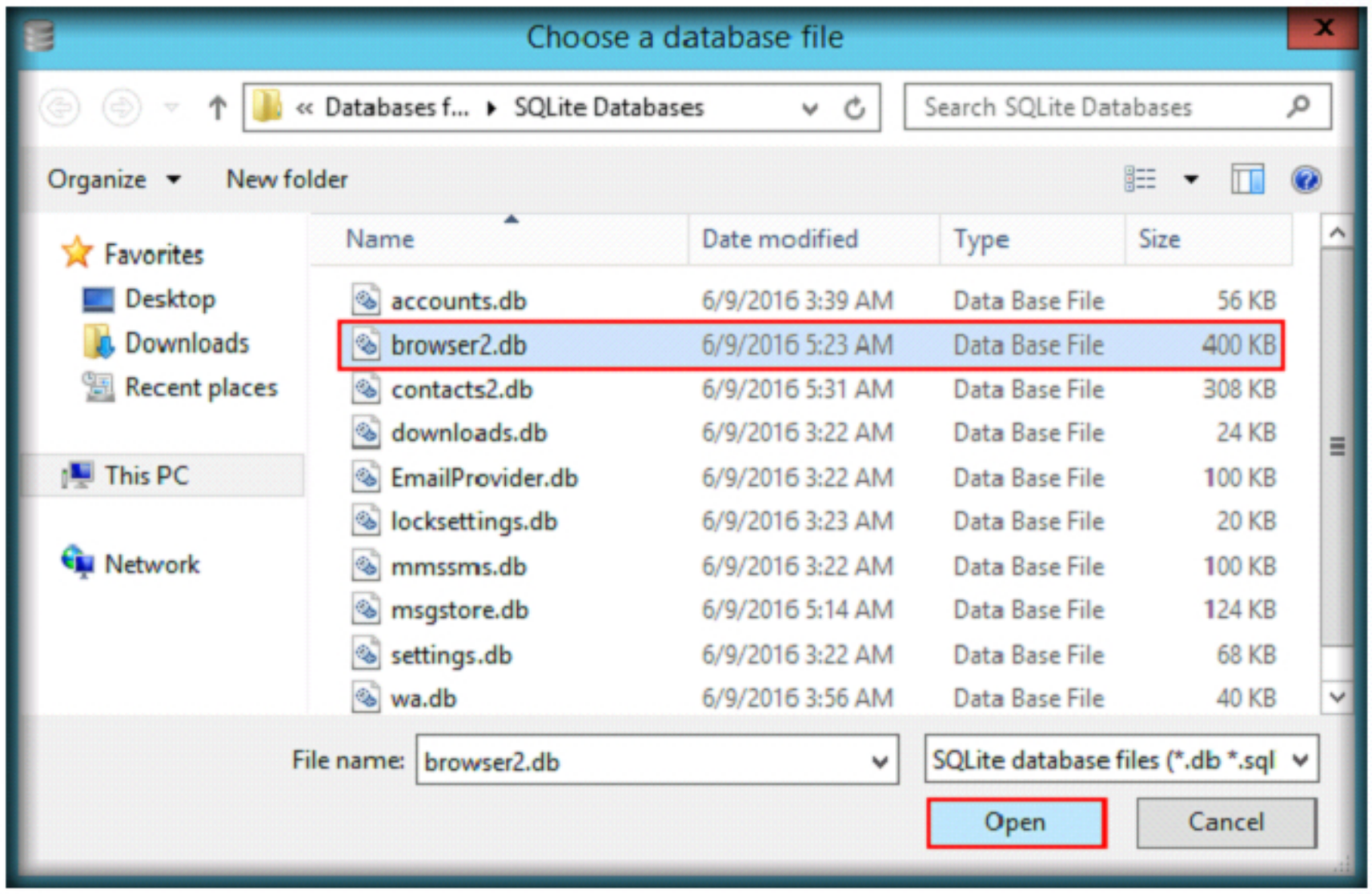

FIGURE 2.8: Navigating to the database files folder

14. Select **Browse Data** tab and select **bookmarks** table from the **Table** drop-down list. This displays all the URLs that were bookmarked on the device as shown in the following screenshot:

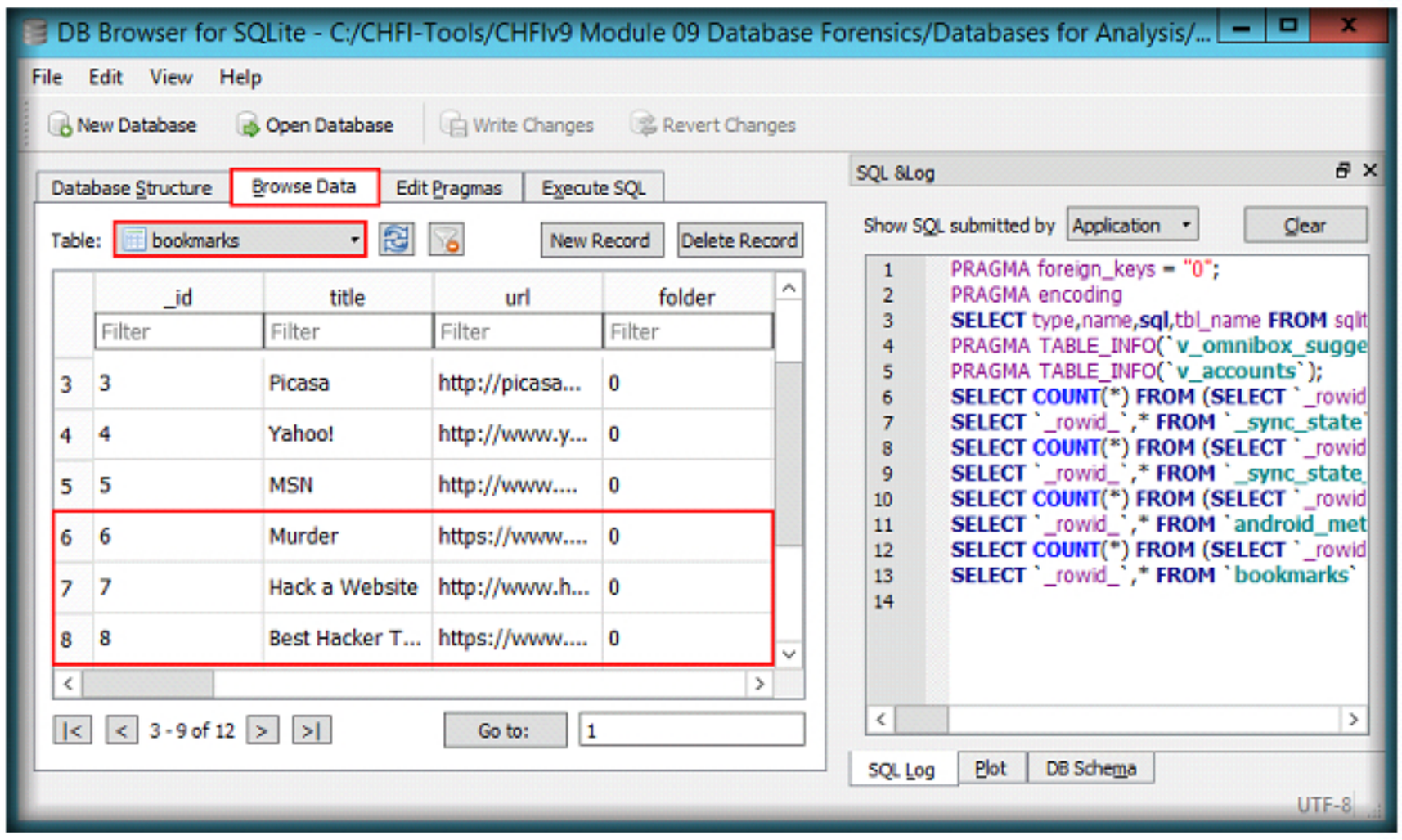

FIGURE 2.9: Viewing all the URLs that were bookmarked on the device

15. Select **history** table from the **Table** drop-down list to view the browser history.

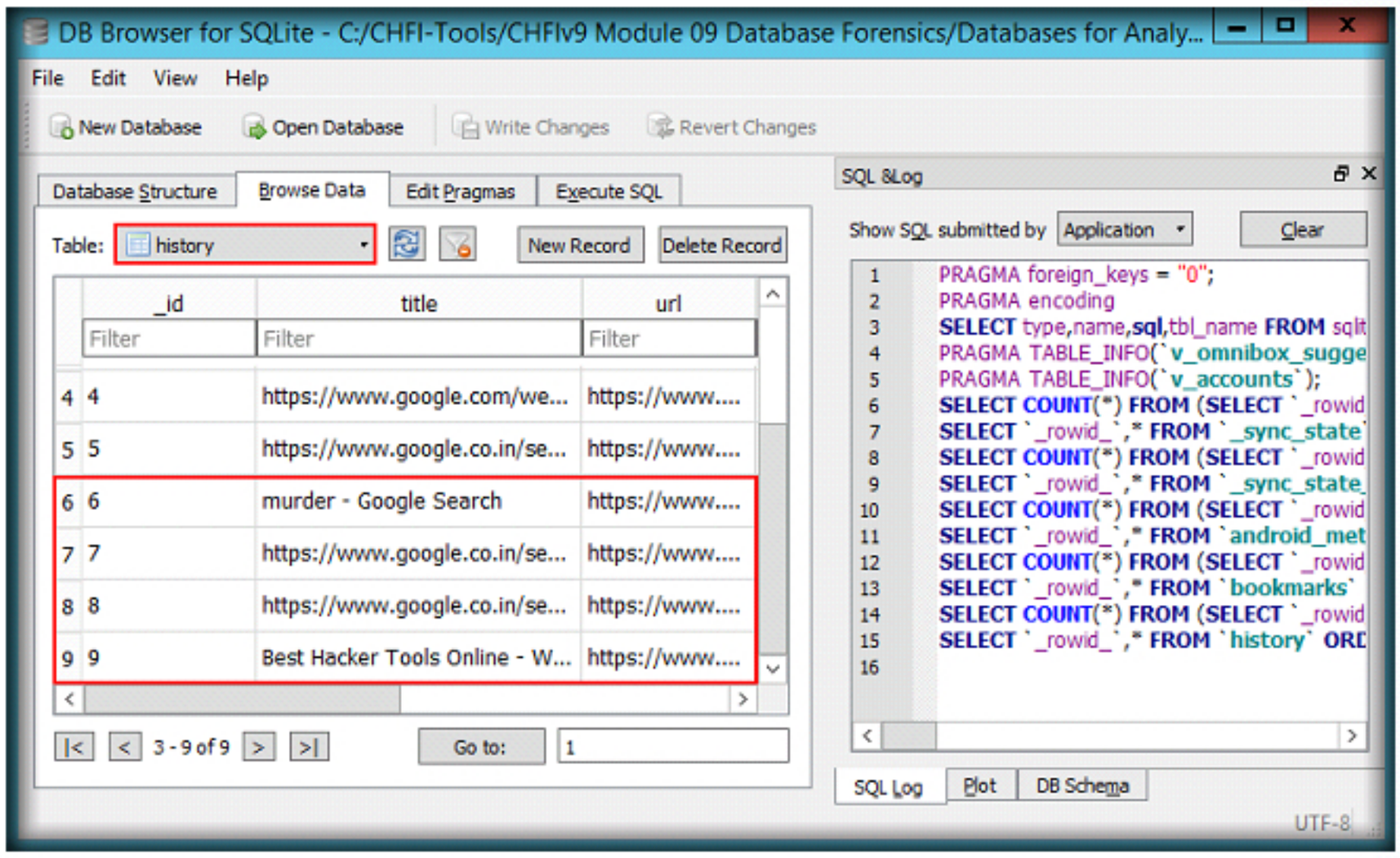

FIGURE 2.10: Viewing browser history

16. The **sqlite_sequence** table stores information related to history (number of websites browsed) and bookmarks (number of websites bookmarked). To view this data, select **sqlite_sequence** table from the **Table** drop-down list.

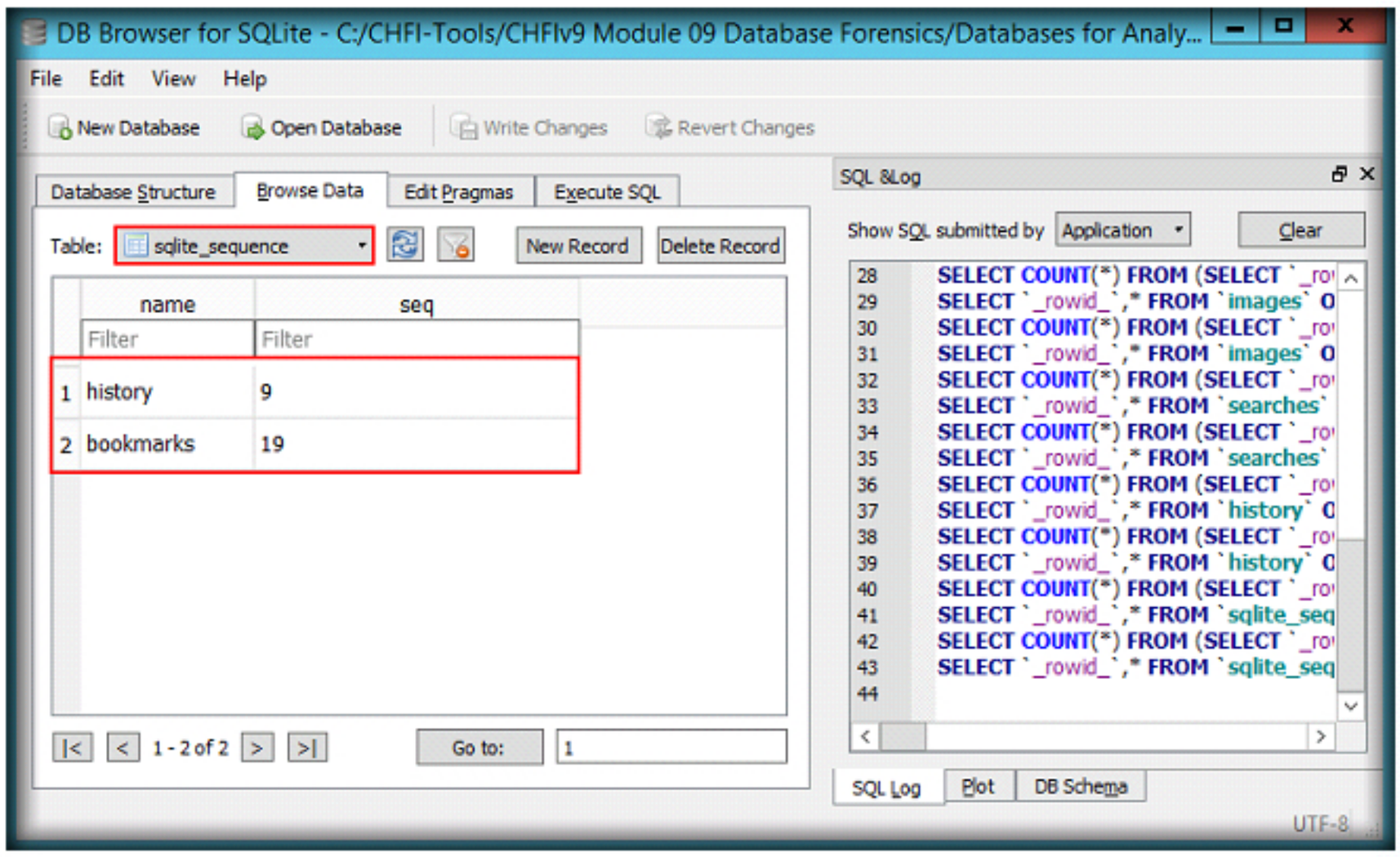

FIGURE 2.11: Viewing the contents sqlite_sequence table

17. Now, we shall examine the contacts database in order to view the contacts in the device and the call history.

18. To view the database, click **Open Database** from the toolbar. **Choose a database file** window appears. Select **contacts2.db** located at **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Databases for Analysis\SQLite Databases**, and click **Open**.

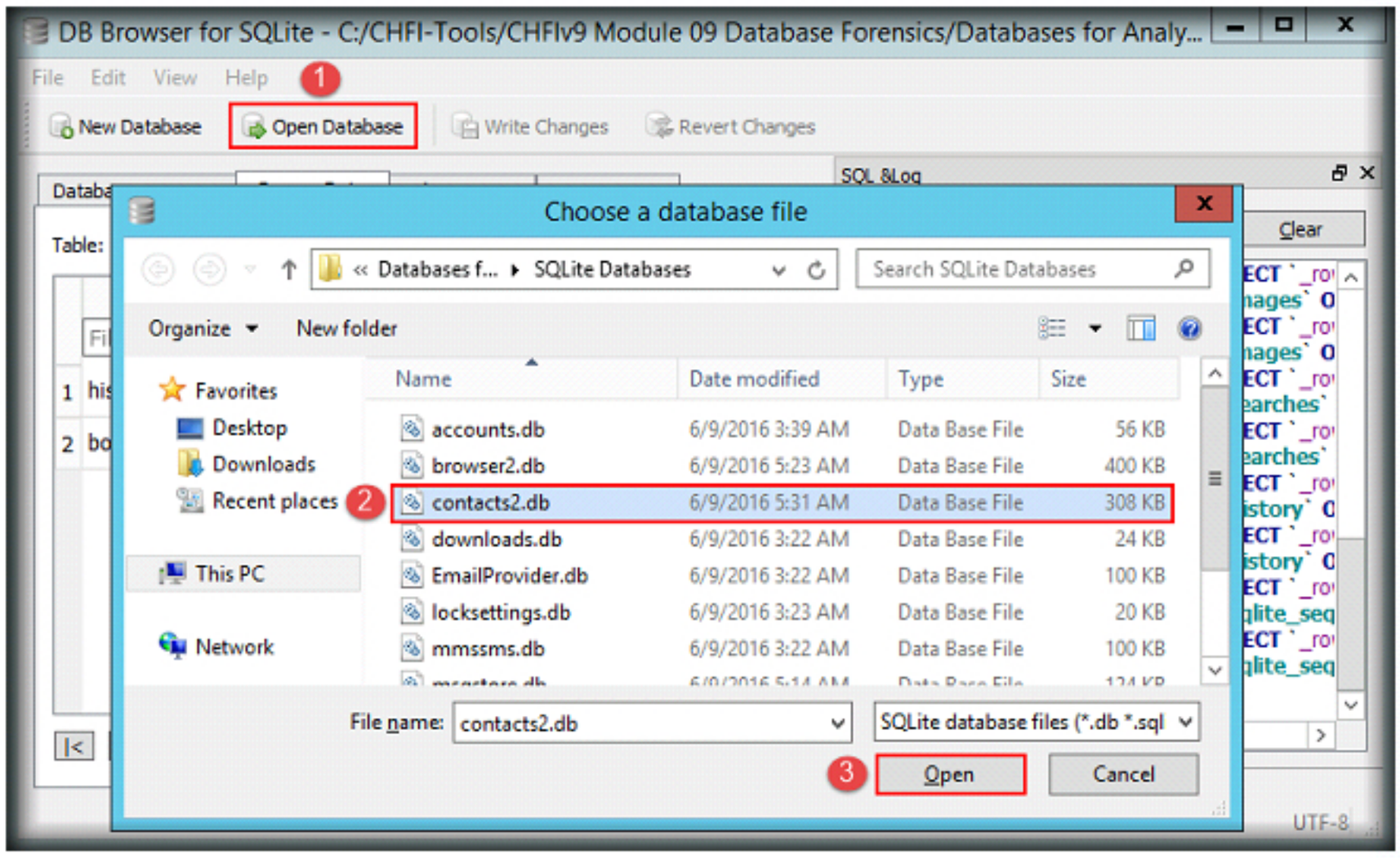

FIGURE 2.12: Navigating to the database files folder

19. If a dialog-box appears stating that a table in the database requires a special collation function, click **Yes** to proceed without the collation.

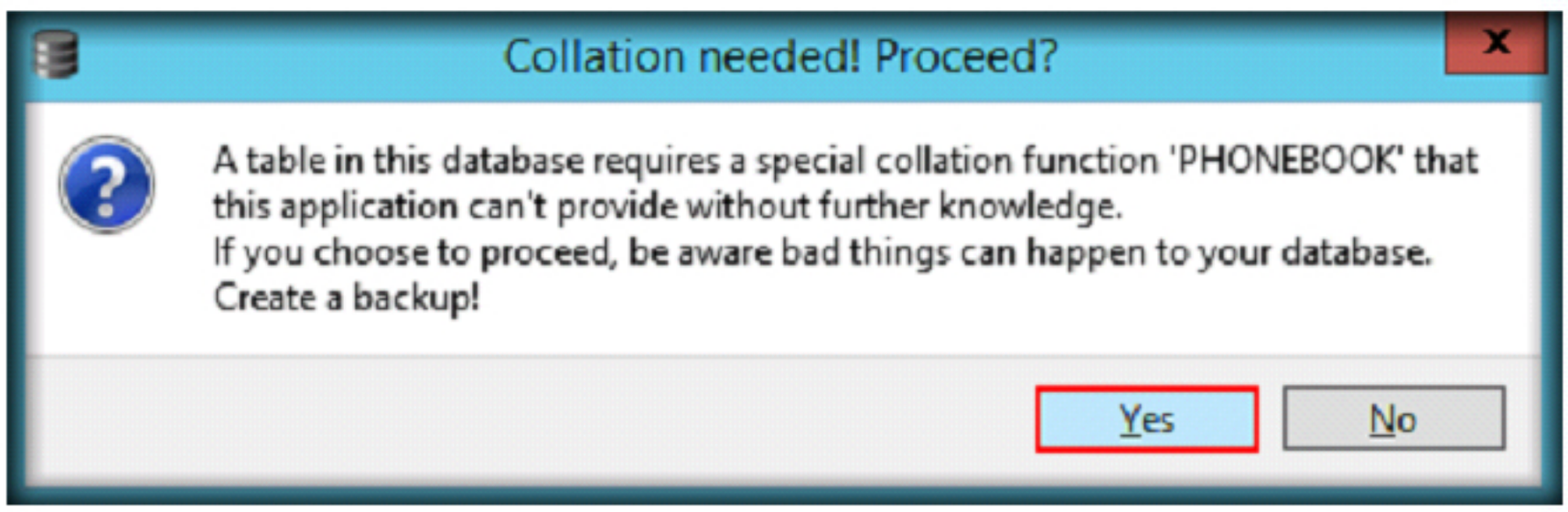

FIGURE 2.13: Collation needed! Proceed? dialog-box

20. The application displays **_sync_state** table by default. To view the contacts stored in the database, select **raw_contacts** table from the **Table** drop-down list. The **raw_contacts** table stores information such as display name, account id, last time contacted, etc.

21. The contents of the table **raw_contacts** are displayed as shown in the following screenshot:

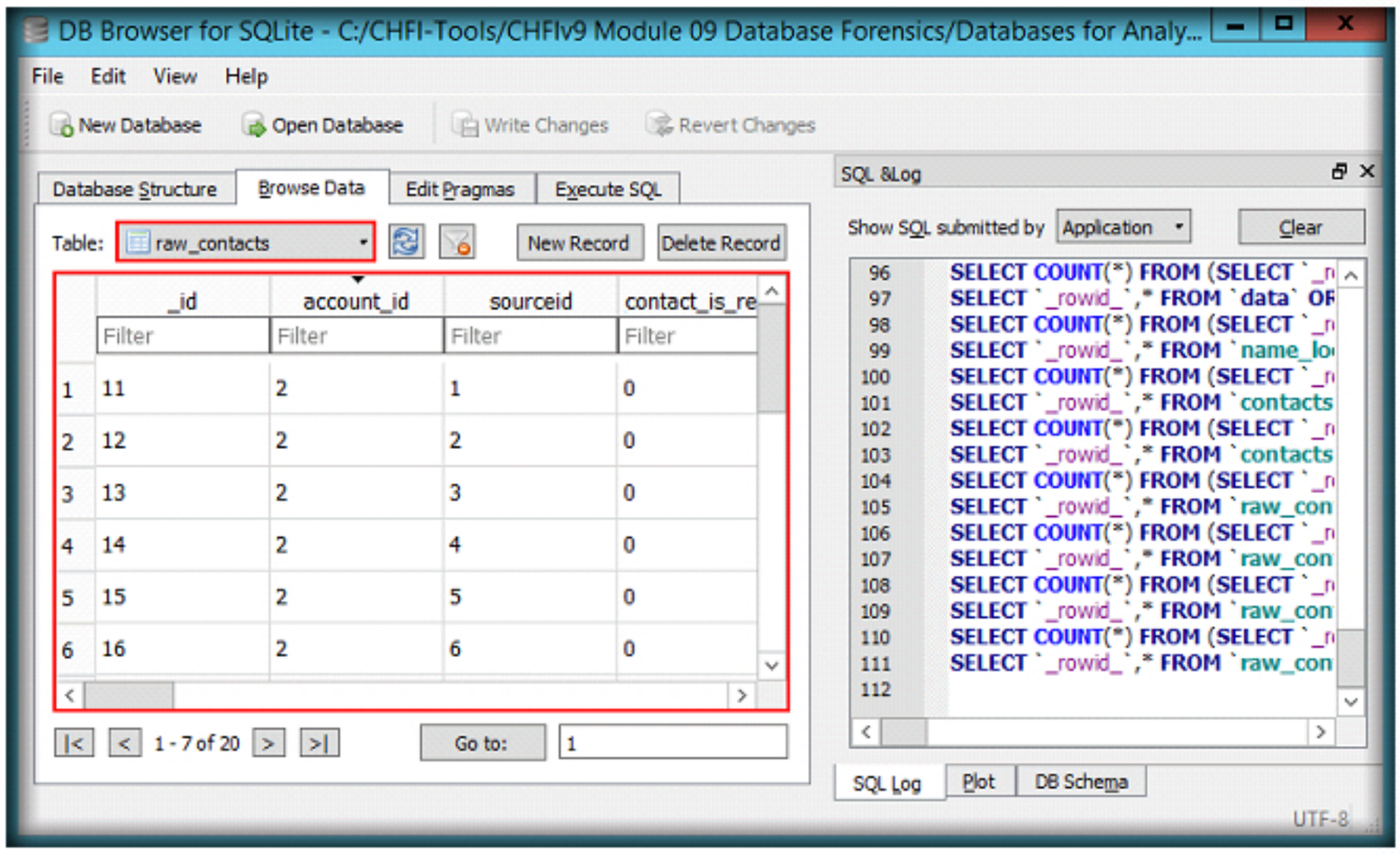

FIGURE 2.14: Viewing contents of raw_contacts table

22. You may scroll down and scroll to the right of the table to view the data stored in the table.

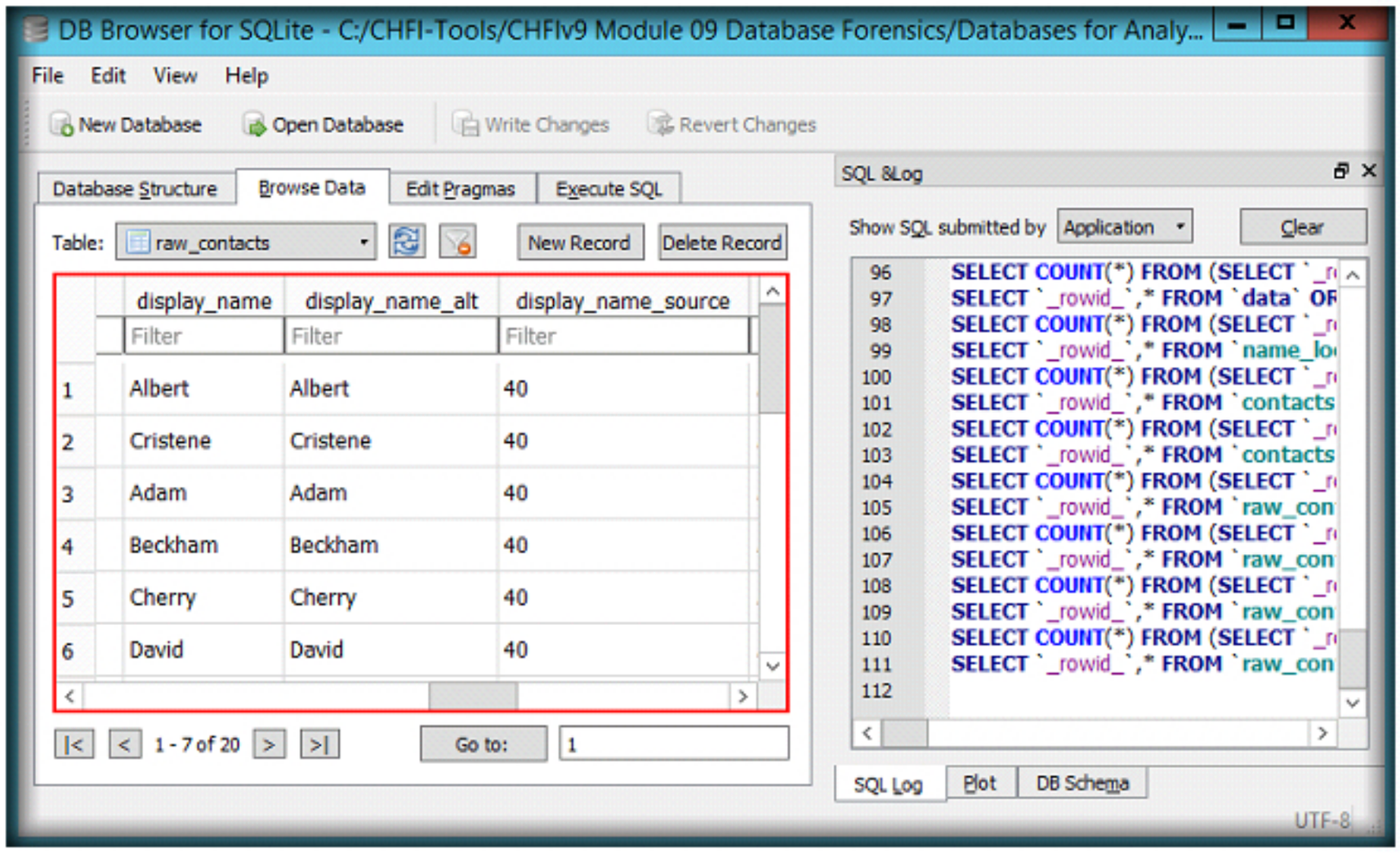

FIGURE 2.15: Viewing contents of raw_contacts table

23. The **calls** table contains the call history associated with the device. This table contains details such as the dialed numbers, dialed contact name, timestamp, call duration, etc.
24. To view this information, select **calls** from the **Table** drop-down list.

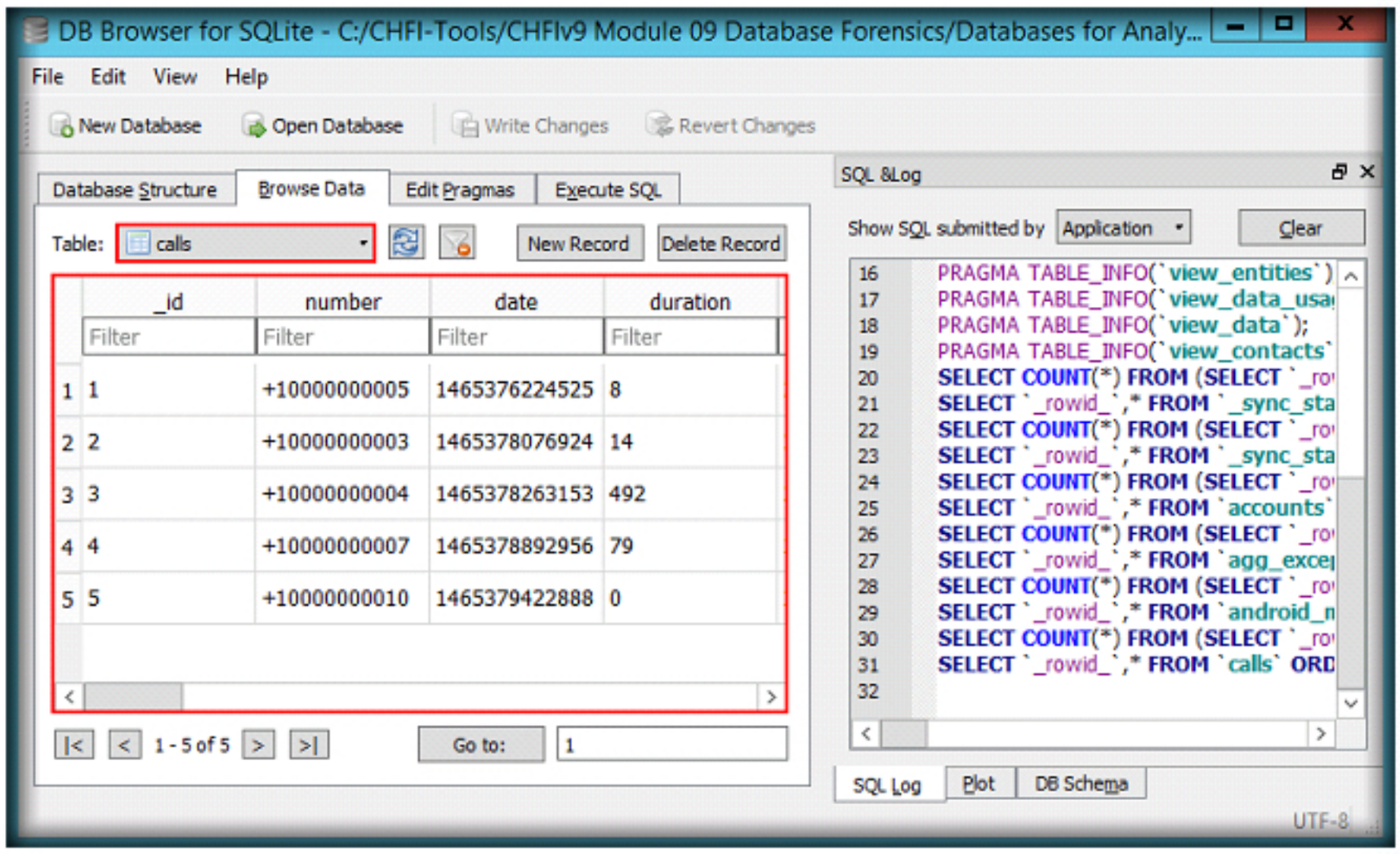

FIGURE 2.16: Viewing contents of calls table

25. You may scroll down and scroll to the right of the table to view the data stored in the table.

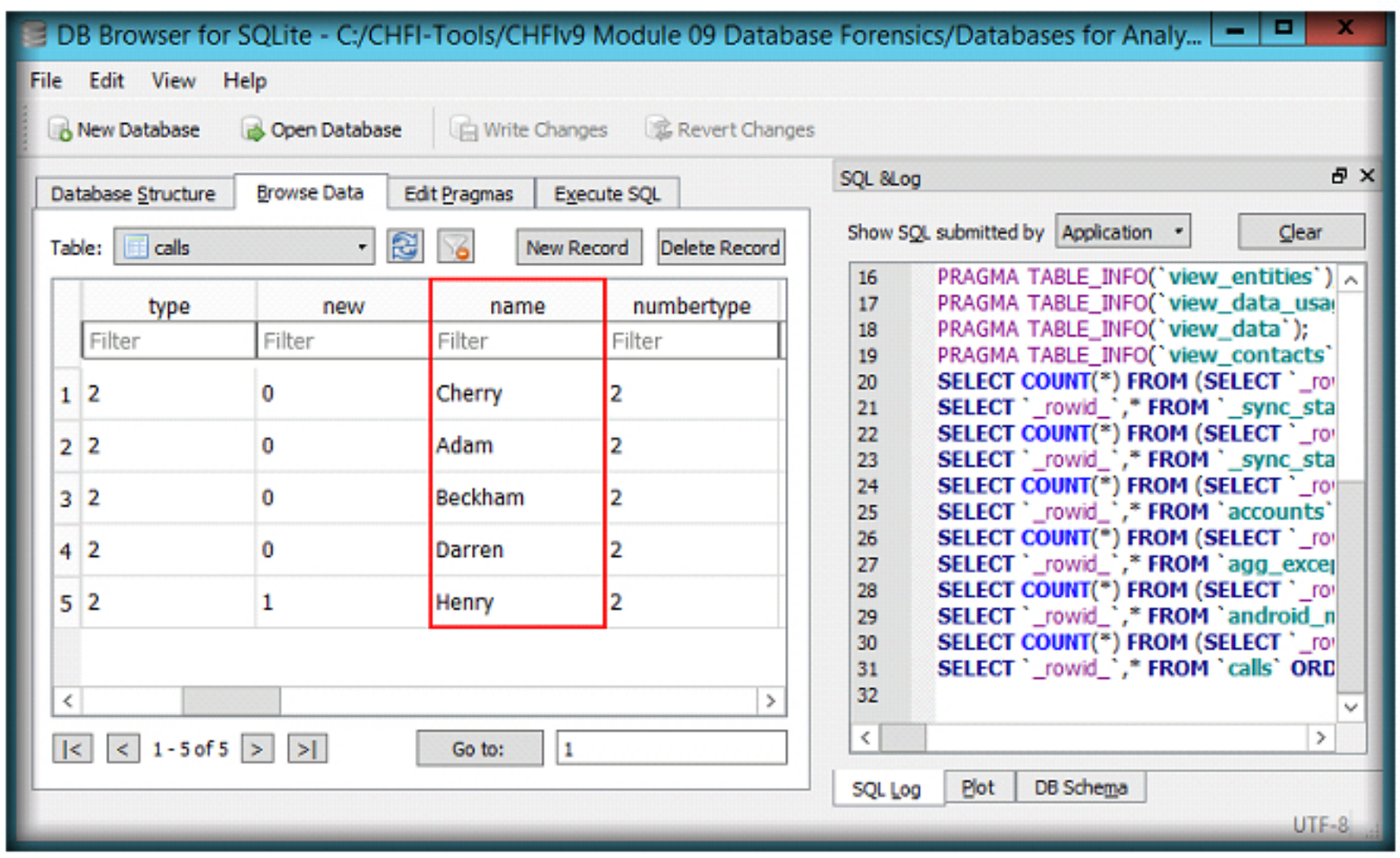

FIGURE 2.17: Viewing contents of calls table

26. Now, we shall view the data stored in **msgstore** database. The **msgstore** database contains information related to the messages stored on the device, timestamps of the sent and received messages, subject of the message, etc.
27. To view this database, click **Open Database** from the toolbar. **Choose a database file** window appears. Select **msgstore.db** located at **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Databases for Analysis\SQLite Databases**, and click **Open**.

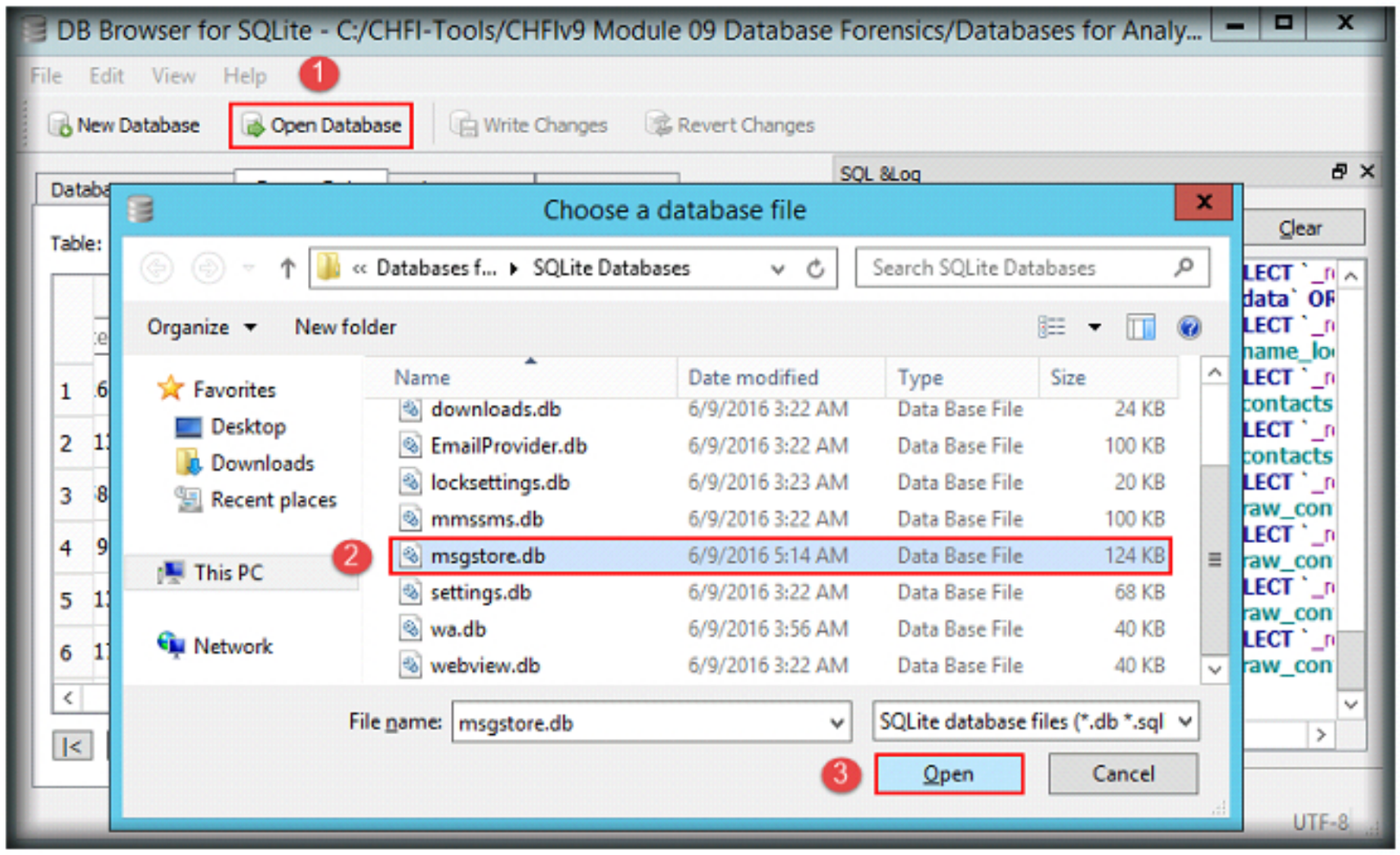

FIGURE 2.18: Navigating to the database files folder

28. Select **chat_list** from the **Table** drop-down list. The **chat_list** table contains information such as subject of the message, key remote id, message creation time, etc., as shown in the following screenshot:

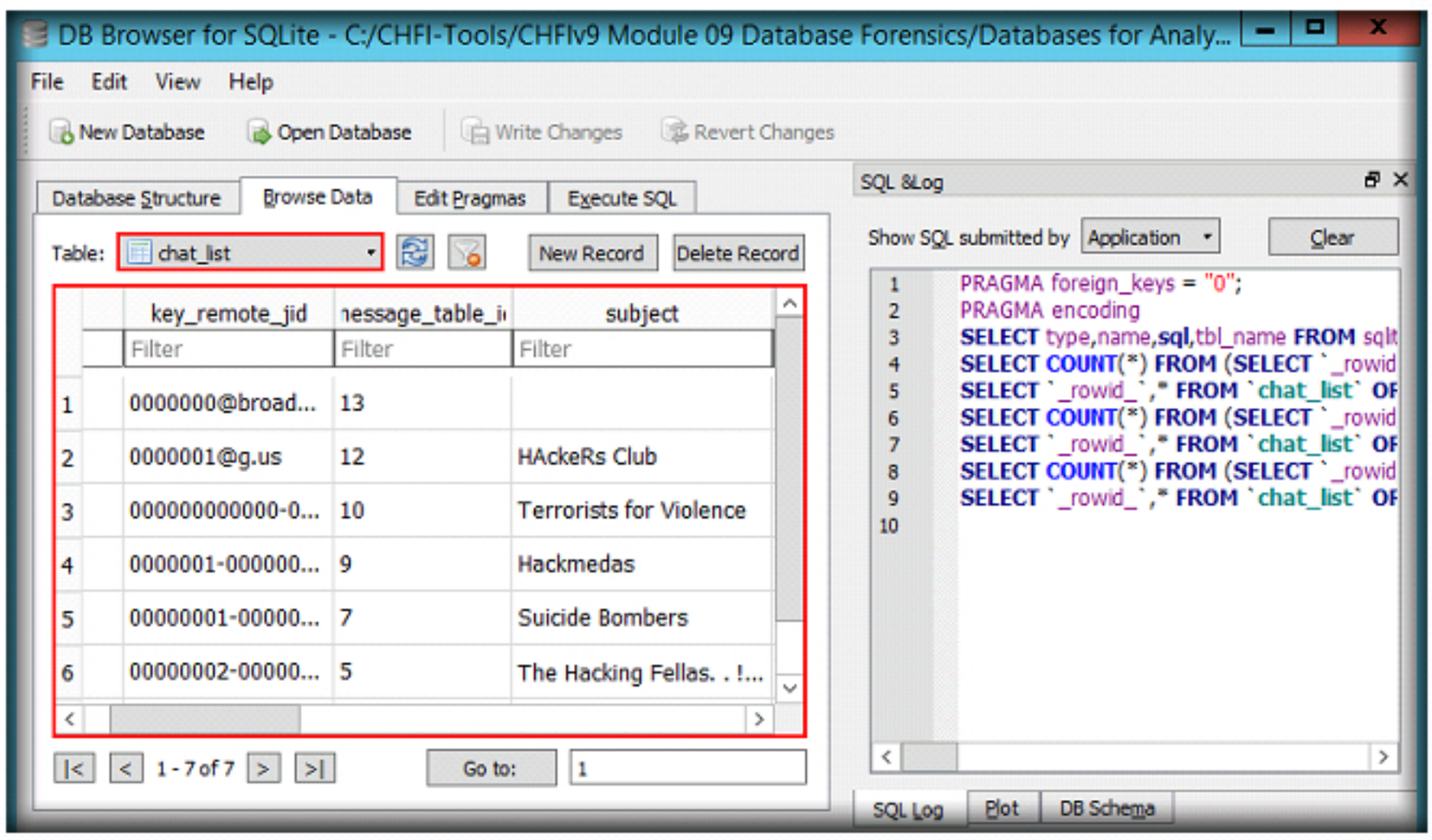

FIGURE 2.19: Viewing contents of chat_list table

29. In the same way, you may analyze the other tables in the database in order to find more information associated with the database.

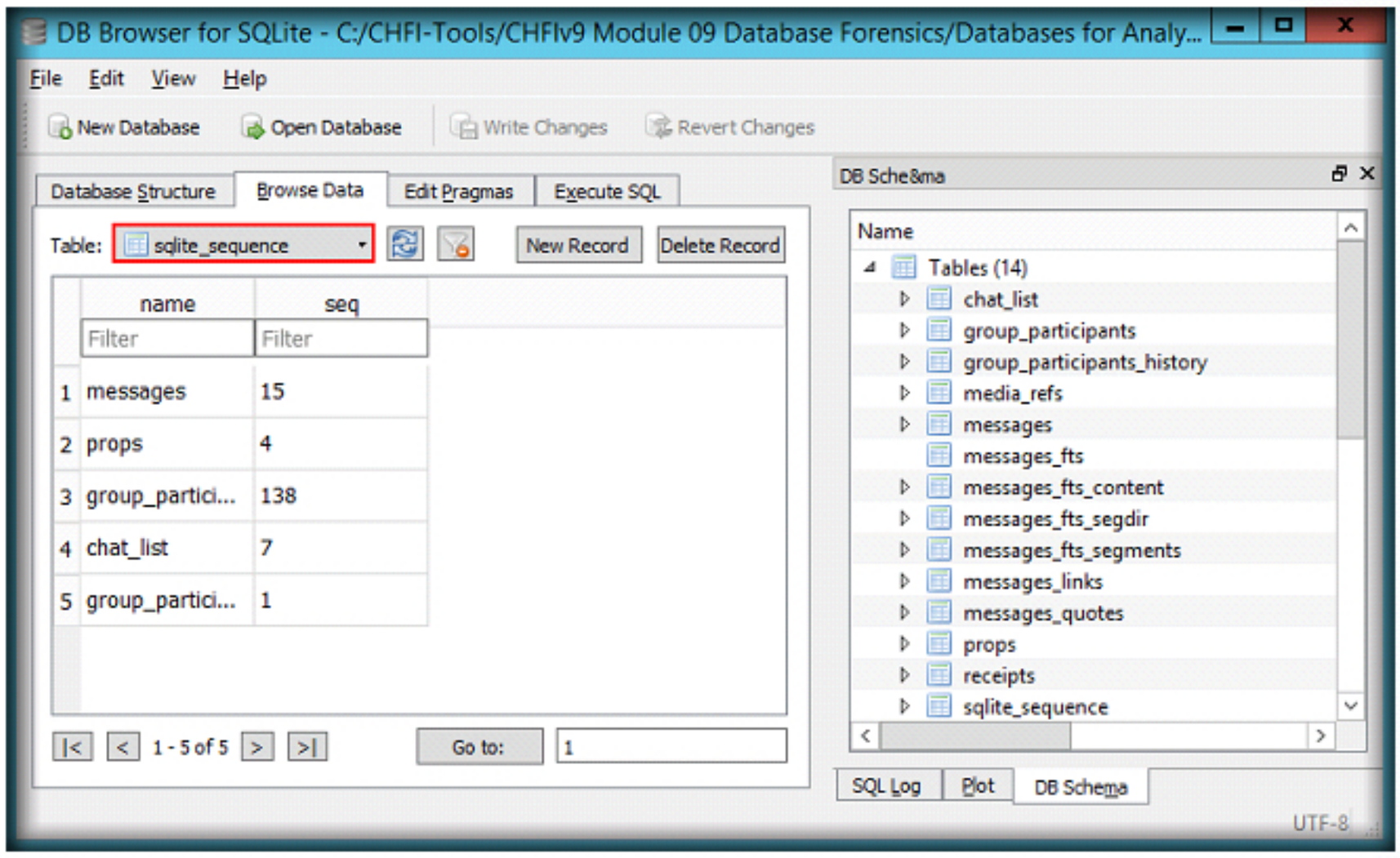

FIGURE 2.20: Viewing the contents of sqlite_sequence table

30. Now, we shall view the data stored in **WhatsApp** database. The **wa** database contains information related to the WhatsApp messages stored on the device, timestamps of the sent and received messages, subject of the message, etc.

31. To view this information, click **Open Database** from the toolbar. **Choose a database file** window appears. Select **wa.db** located at **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Databases for Analysis\SQLite Databases**, and click **Open**.

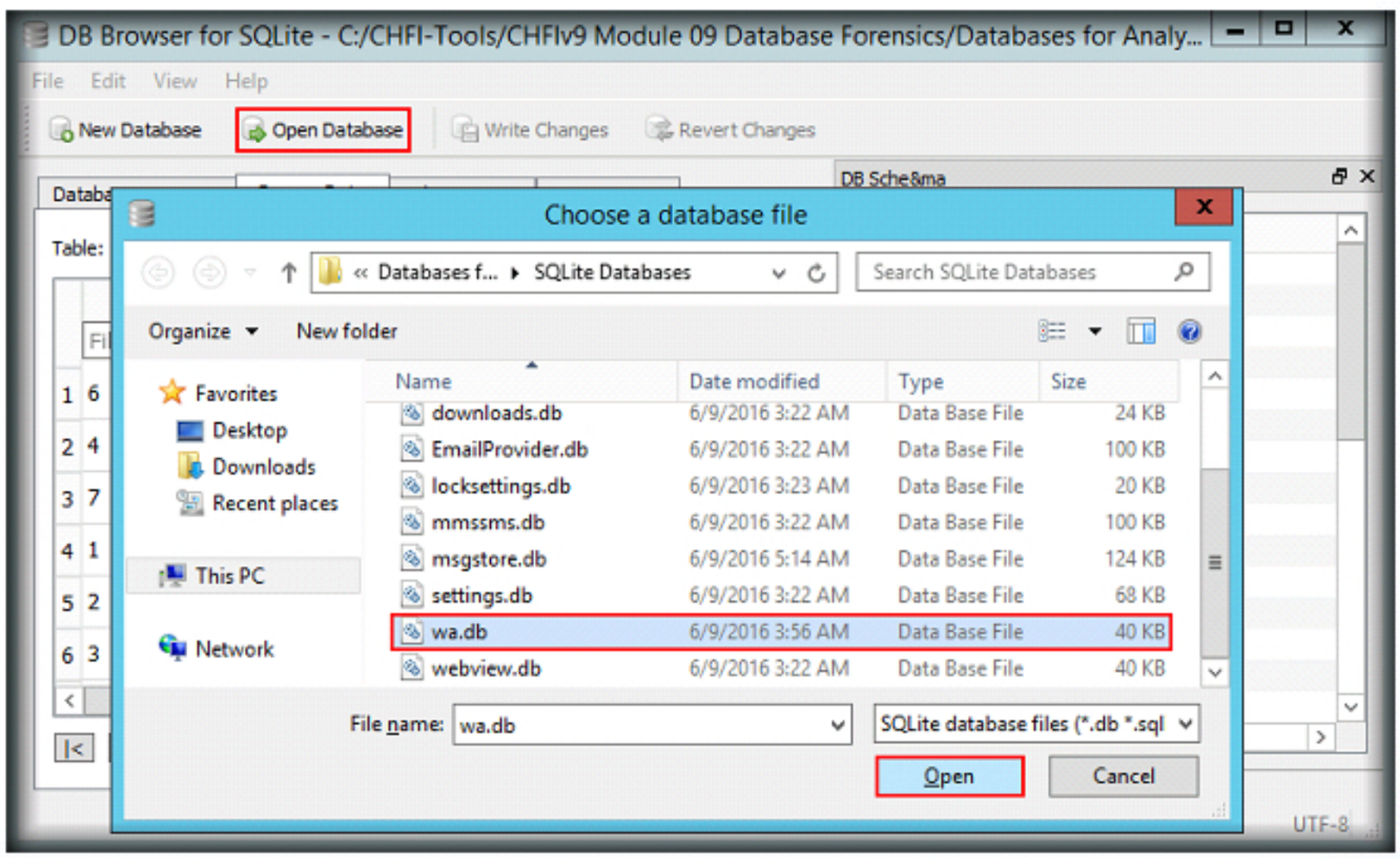

FIGURE 2.21: Navigating to the database files folder

32. You may browse various tables in the database to view information such as number of WhatsApp contacts, WhatsApp contacts' names, etc. as shown in the following screenshots:

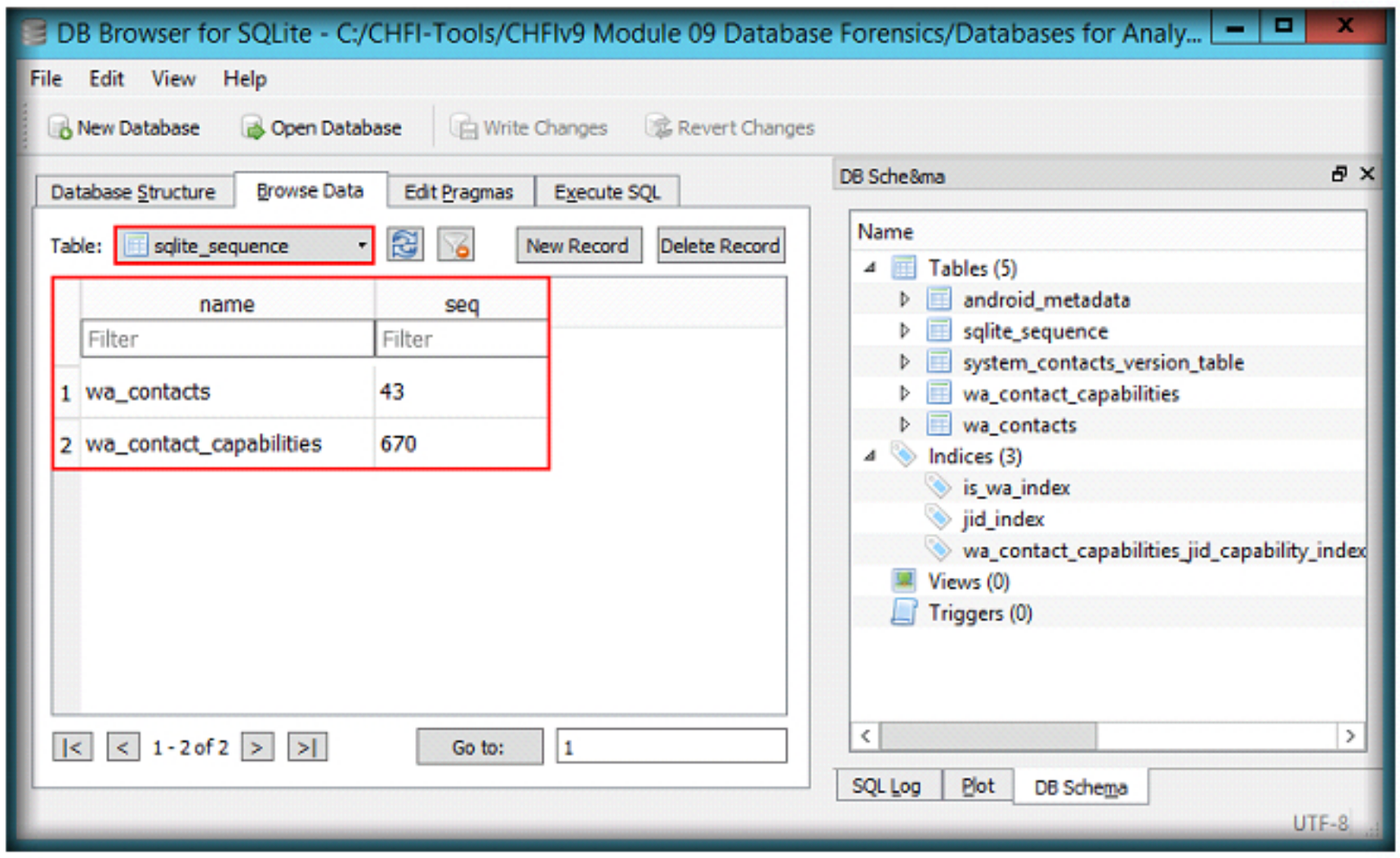

FIGURE 2.22: Viewing data stored in WhatsApp database

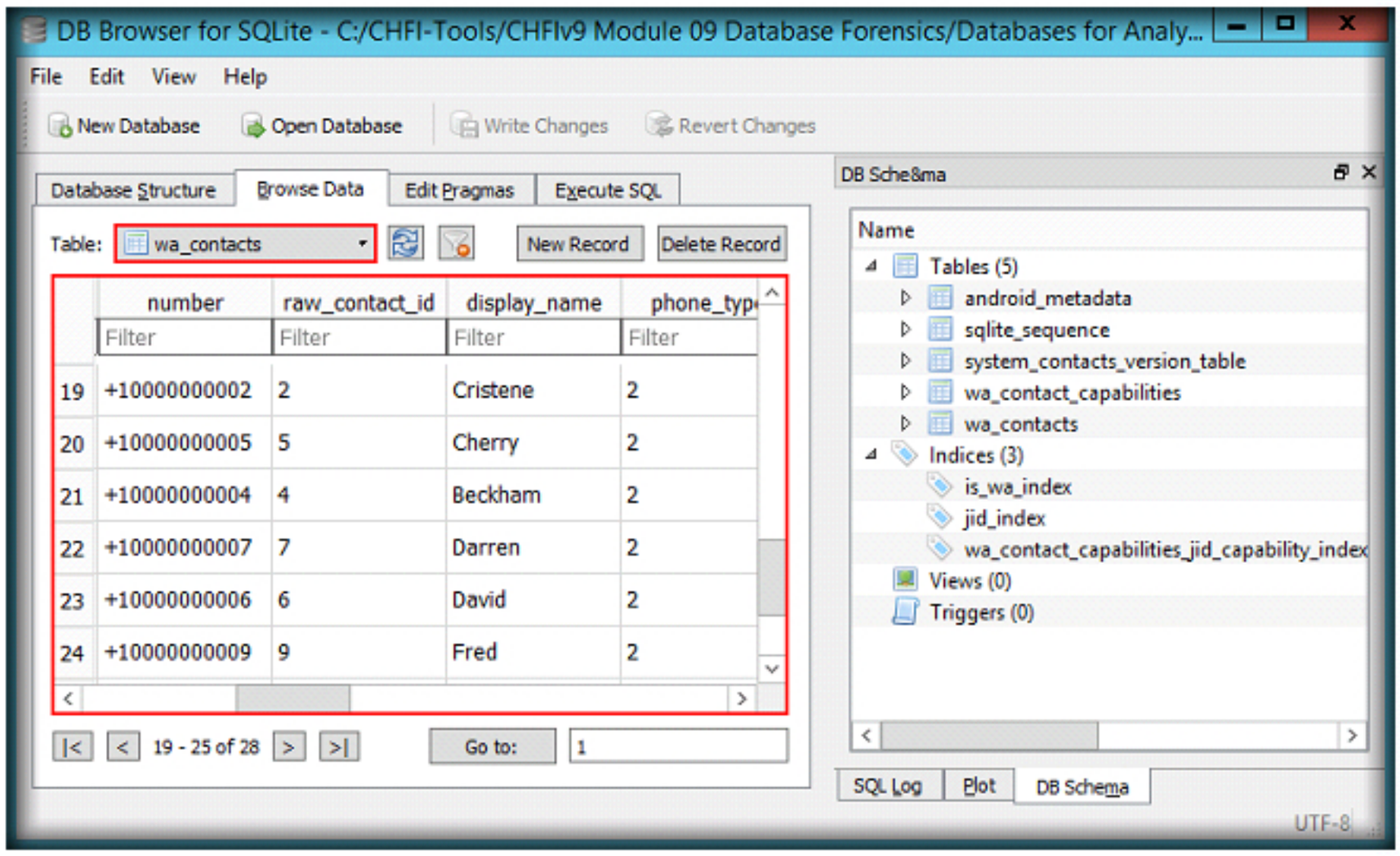

FIGURE 2.23: Viewing data stored in WhatsApp database

33. The **locksettings** database contains the settings such as the status of the lock screen, lockscreen password type, status of the lockscreen pattern autolock (enabled or disabled), visibility of the lockscreen pattern, etc.

34. To view the settings, click **Open Database** from the toolbar. **Choose a database file** window appears. Select **locksettings.db** located at **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Databases for Analysis\SQLite Databases**, and click **Open**.

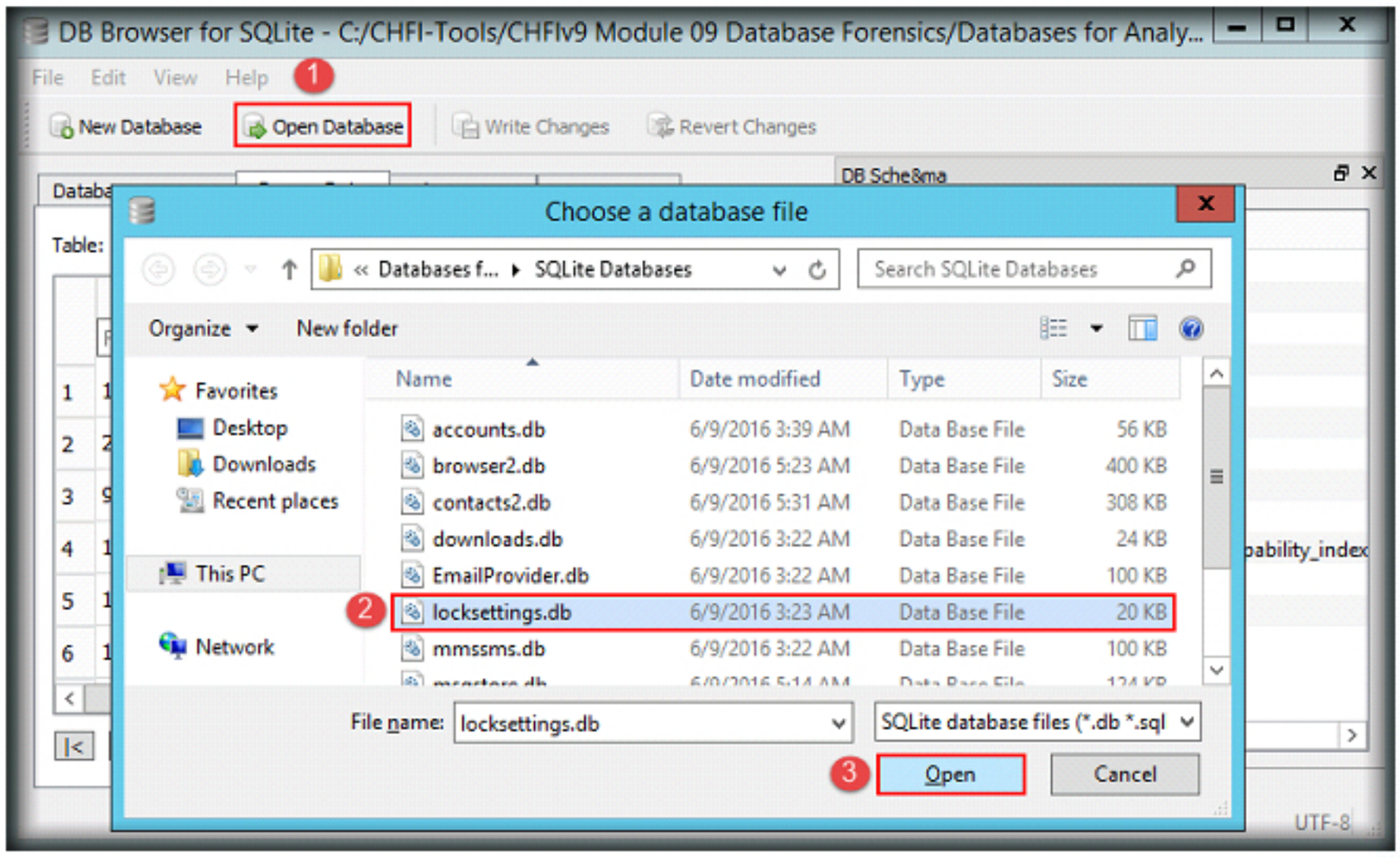

FIGURE 2.24: Navigating to the database files folder

35. Select **locksettings** from the **Table** drop-down list, to view settings associated with the lock screen pattern as shown in the following screenshot:

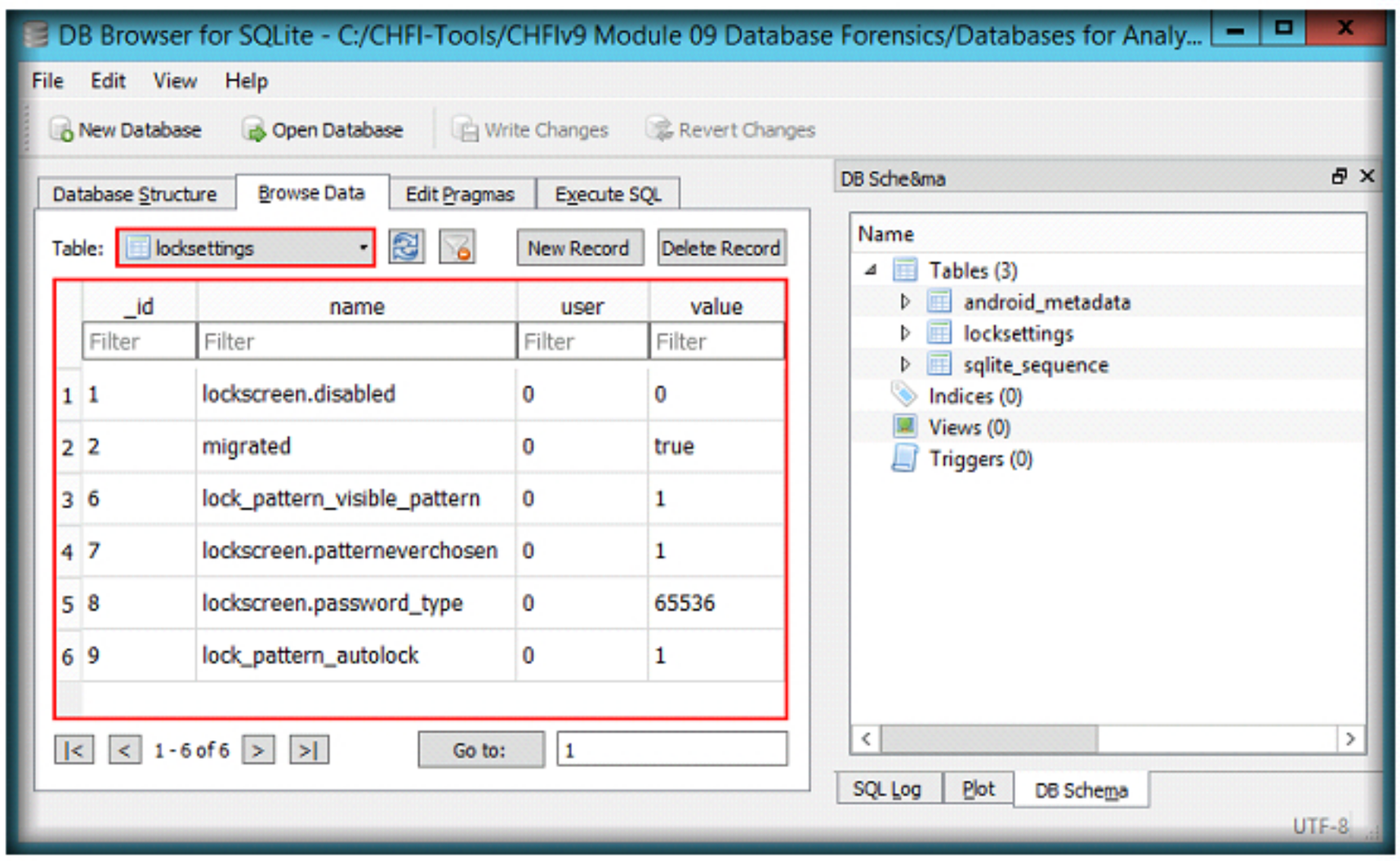

FIGURE 2.25: Viewing the contents of locksettings table

36. This way, as a forensic investigator, you may analyze all the databases that were extracted from the mobile device.

Lab Analysis

Analyze the results and document the findings of the lab.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab

3

Performing Forensic Investigation on a MySQL Server Database

MySQL is an open-source relational database management system for use in web applications.

ICON KEY
Valuable information
Test your knowledge
Web exercise
Workbook review

Lab Scenario

A suspicious post was found on a WordPress website's homepage, which indicates that a suspicious activity has occurred on the backend database. The objective of this lab is to find the malicious user who gained access to **mysql** server and examine the activities performed by him/her on the database named **WordPress**.

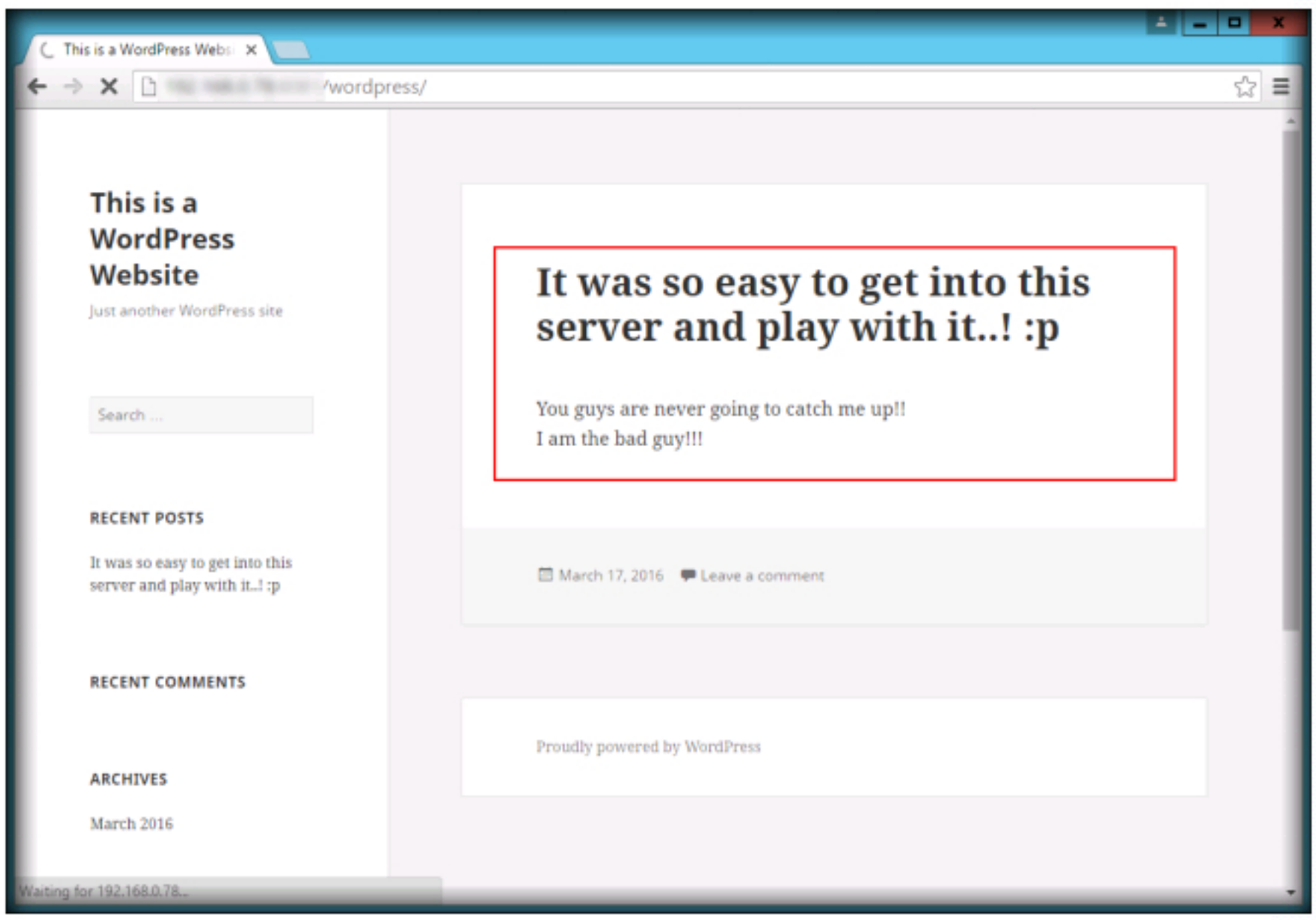

FIGURE 3.1: Examining the suspicious post on a wordpress website's homepage

Lab Objectives

In this lab, you will learn how to examine the databases and find the transactions performed by a suspicious user.

Lab Environment

This lab requires:

- A **Windows Server 2012** virtual Machine
- **Hex Workshop Hex Editor** located at **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Hex Workshop Hex Editor**
- **WampServer** located at **C:\CHFI-Tools\CHFIv9 Lab Prerequisites\WampServer**
- **Microsoft Visual C++ 2012** located at **C:\CHFI-Tools\CHFIv9 Lab Prerequisites\Microsoft Visual C++ 2012**
- Administrative privileges to run the tools

Lab Duration

Time: 20 Minutes

Overview of the Lab

- Install the **WampServer** and understand its working.
- Use **Hex Workshop Hex Editor** to find the hidden evidence files from the database.

Lab Tasks

TASK 1

Install Hex Workshop Hex Editor

1. Before beginning this lab, you need to logon to **Windows Server 2012** virtual machine.
2. Navigate to **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Hex Workshop Hex Editor**, double-click **hw_v680.exe**.

3. Follow the wizard-driven installation steps to install the application.

Note: If **Open File - Security Warning** pop-up appears, click **Run**.

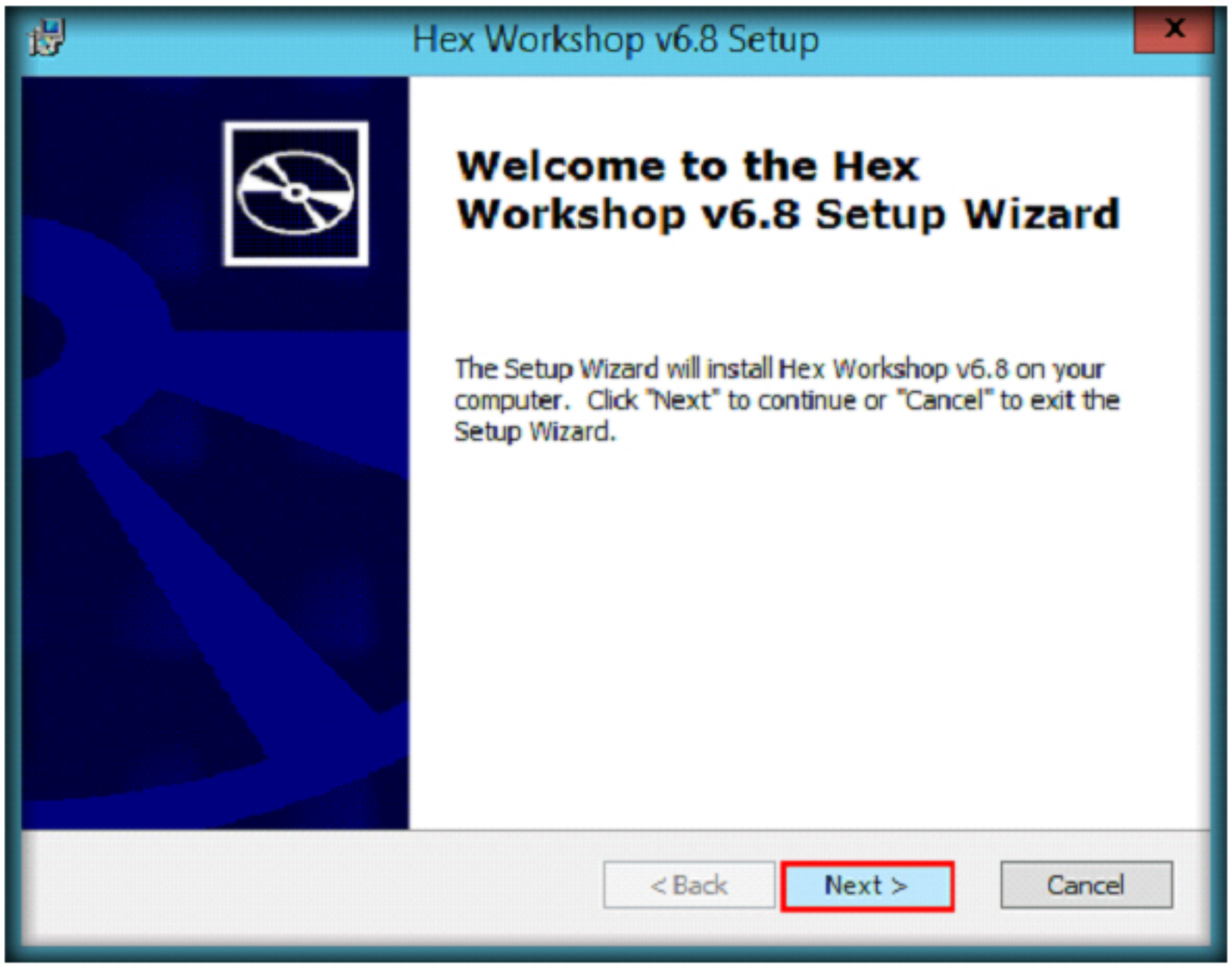

FIGURE 3.2: Hex Workshop Hex Editor installation wizard

4. On completion of installation, if an **Installer Information** dialog-box appears, click **Yes** to restart the virtual machine.

5. On installing the applications, navigate to **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Evidences\data**, copy **wordpress_evidence.sql** and paste it in **C:\wamp\bin\mysql\mysql5.6.17\bin**.

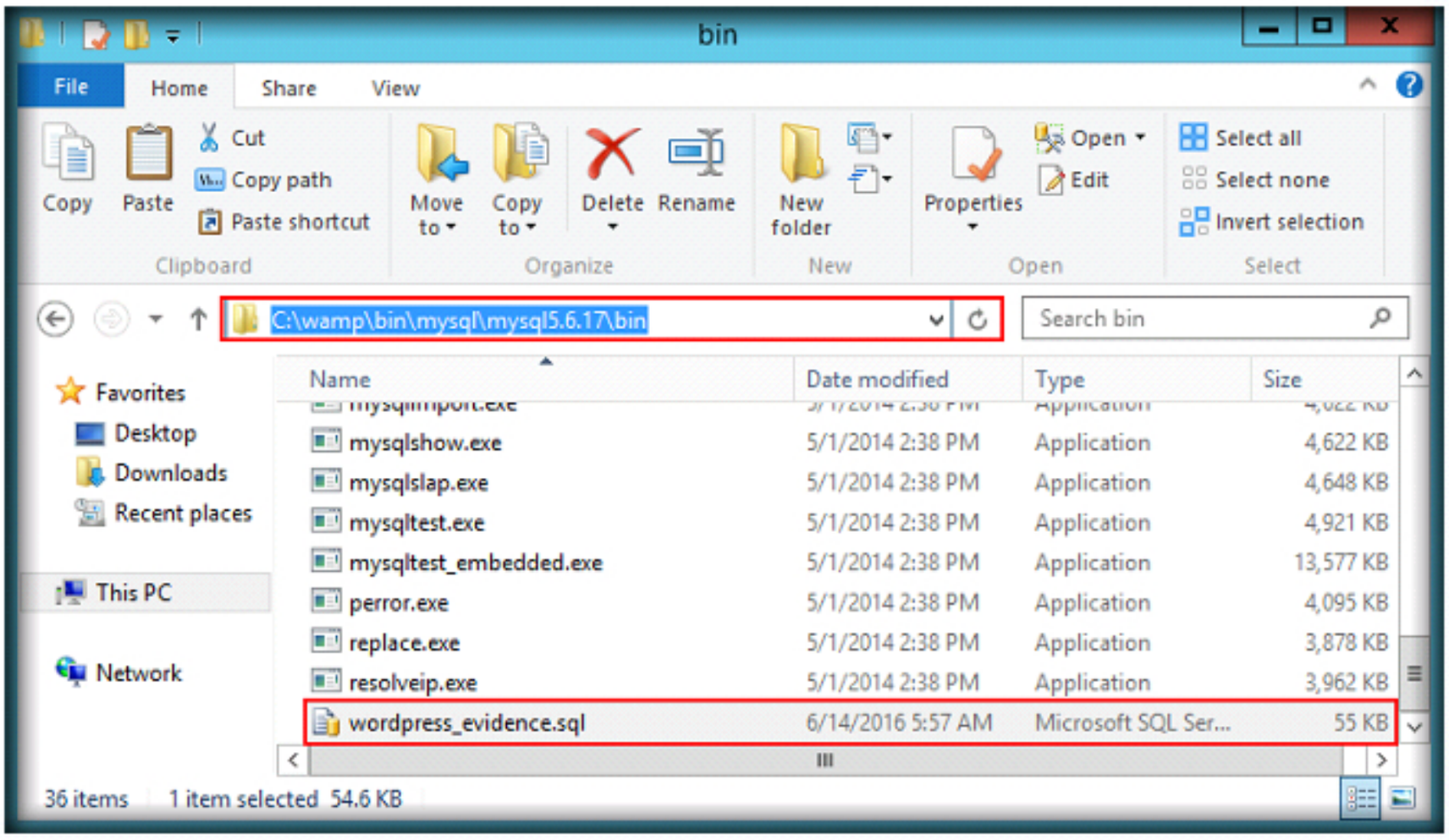

FIGURE 3.3: Copying and pasting the evidence file in the required folder

6. Now, navigate to **C:\wamp\bin\mysql\mysql5.6.17**, select **bin** folder, press **Shift** on the keyboard and right-click on the folder. Context menu appears. Select **Open command window here**.

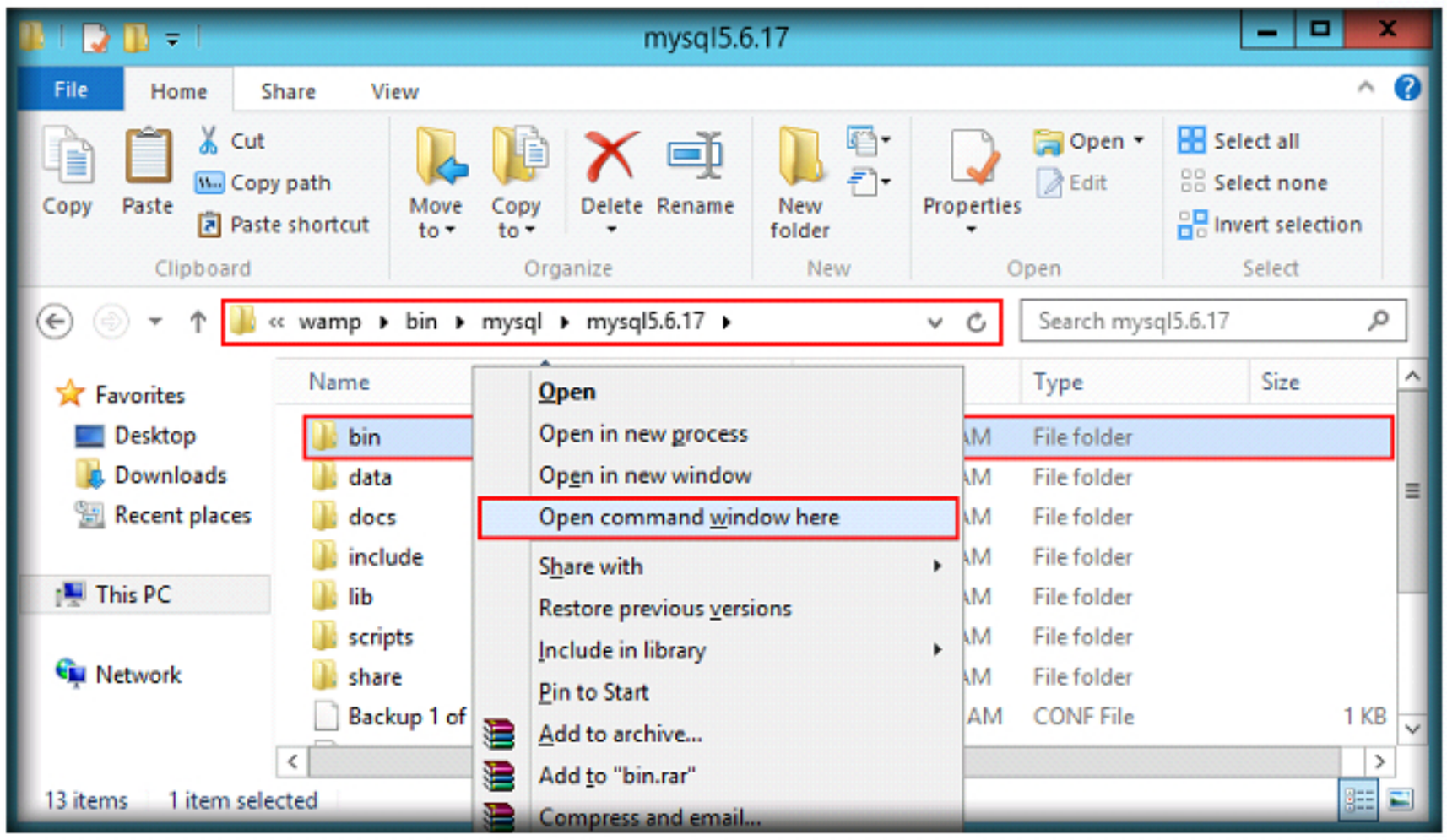

FIGURE 3.4: Selecting Open command window here option from the context menu

7. Command prompt appears. Point the location of **bin** folder. Type **mysql -u root -p** and press **Enter**. You will be asked to enter password. In the **Enter password** field, press **Enter** without issuing any password.

8. A mysql shell appears as shown in the following screenshot:

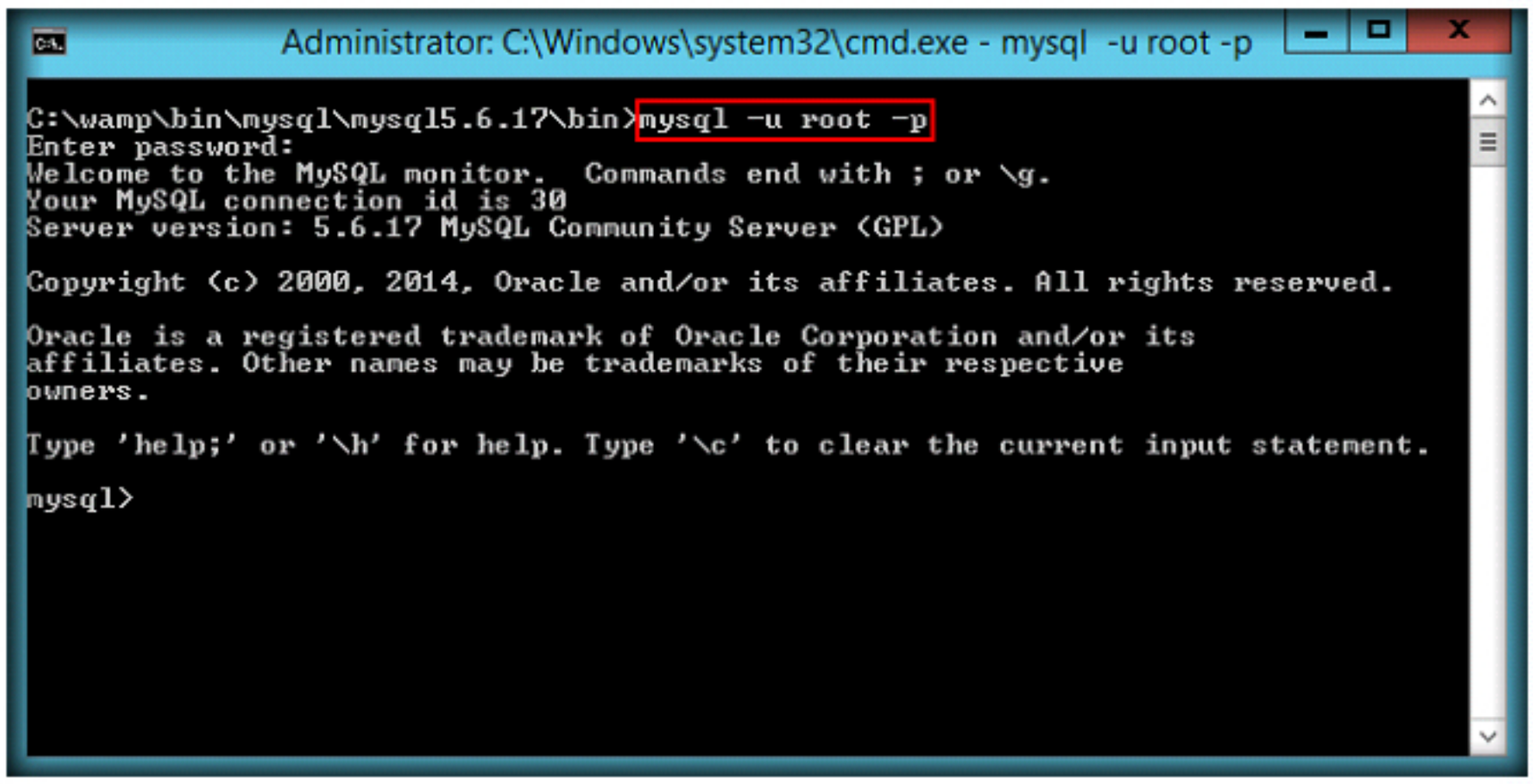

FIGURE 3.5: mysql shell

9. Type **create database wordpress;** in the shell and press **Enter**. This creates a database named **wordpress**. Once done, type **\q** and press **Enter** to come out of the mysql shell.

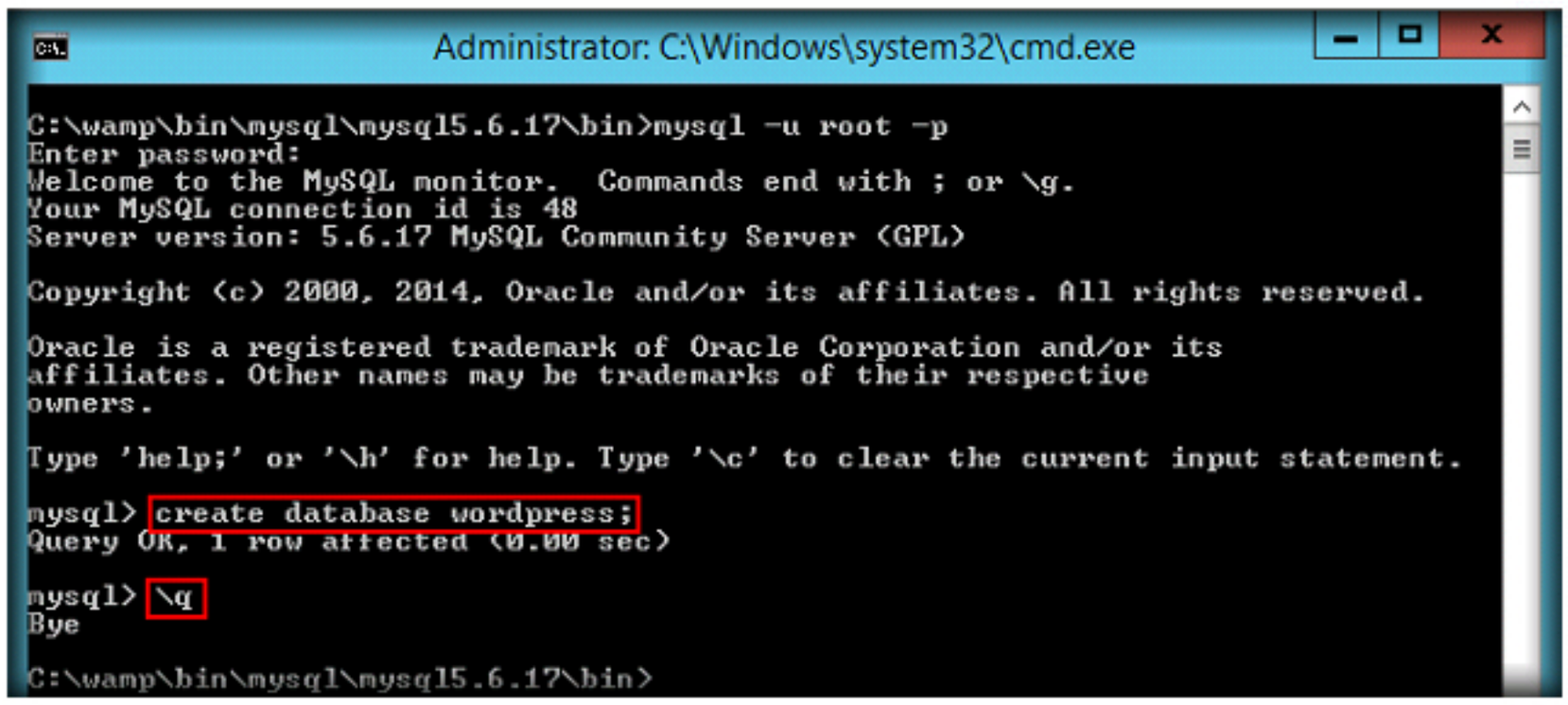

FIGURE 3.6: Creation of database wordpress

10. Now, we shall copy all the contents of the dump file to the newly created database.
11. To copy, type **mysql -u root -p wordpress < wordpress_evidence.sql** in the command prompt and press **Enter**.
12. You will be asked to enter password. In the **Enter password** field, press **Enter** without issuing any password.

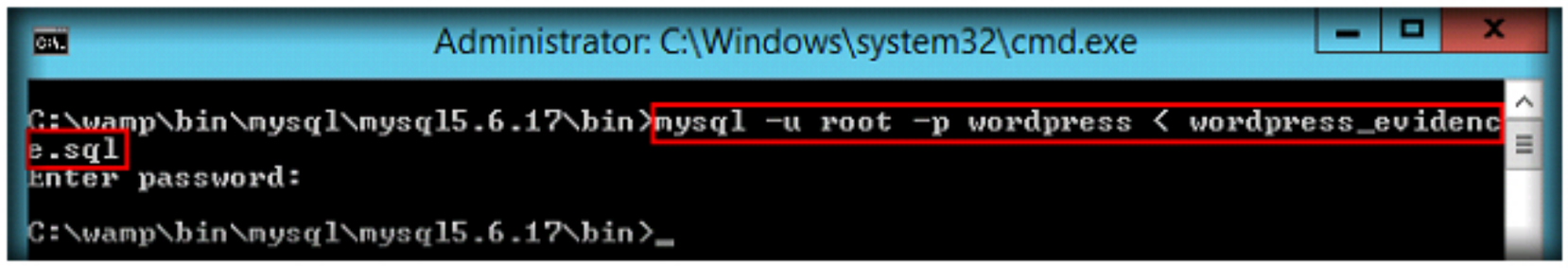

FIGURE 3.7: Copying contents of dump file

13. Once the backup is copied to the database, we shall login to mysql shell and (by entering **mysql -u root -p** and then issuing **empty password**) start examining the database. To examine the database, we need to use the database.
14. Type **use wordpress;** and press **Enter** to use the **wordpress** database.

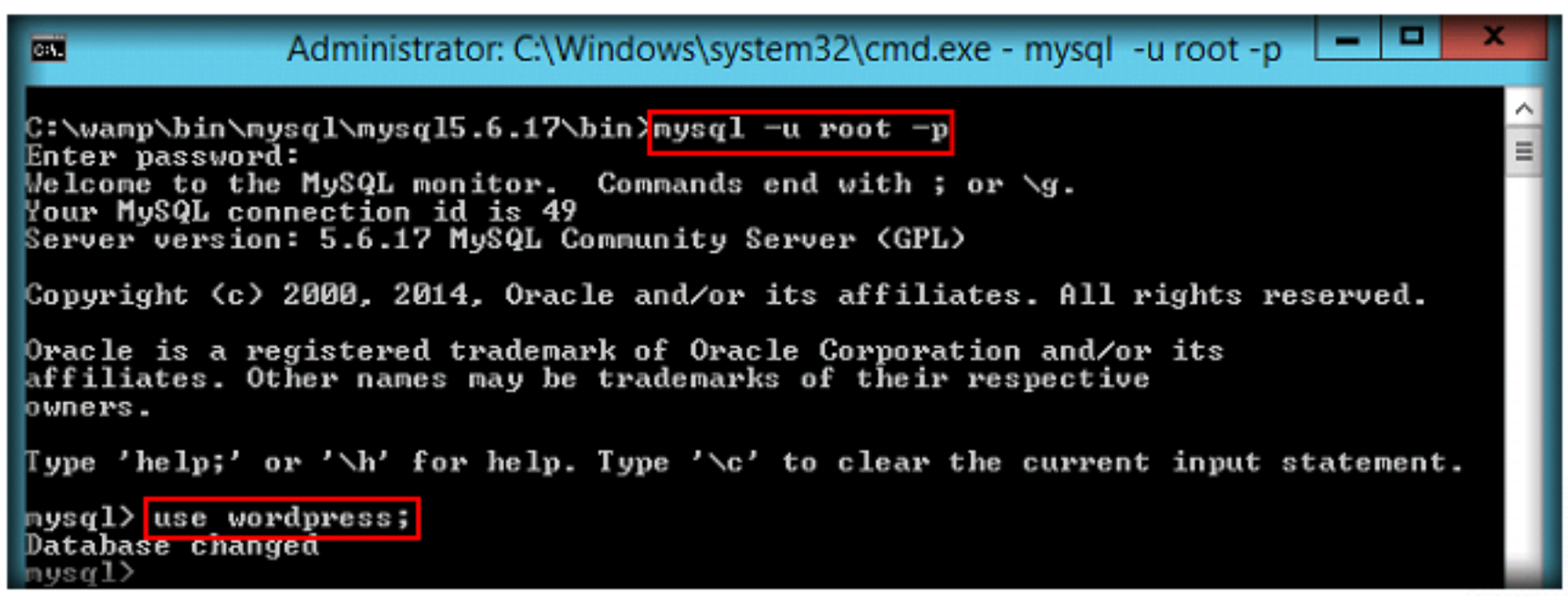

FIGURE 3.8: Examining the database

15. Now, we shall view the tables in this database. To view, type **show tables;** and press **Enter**.

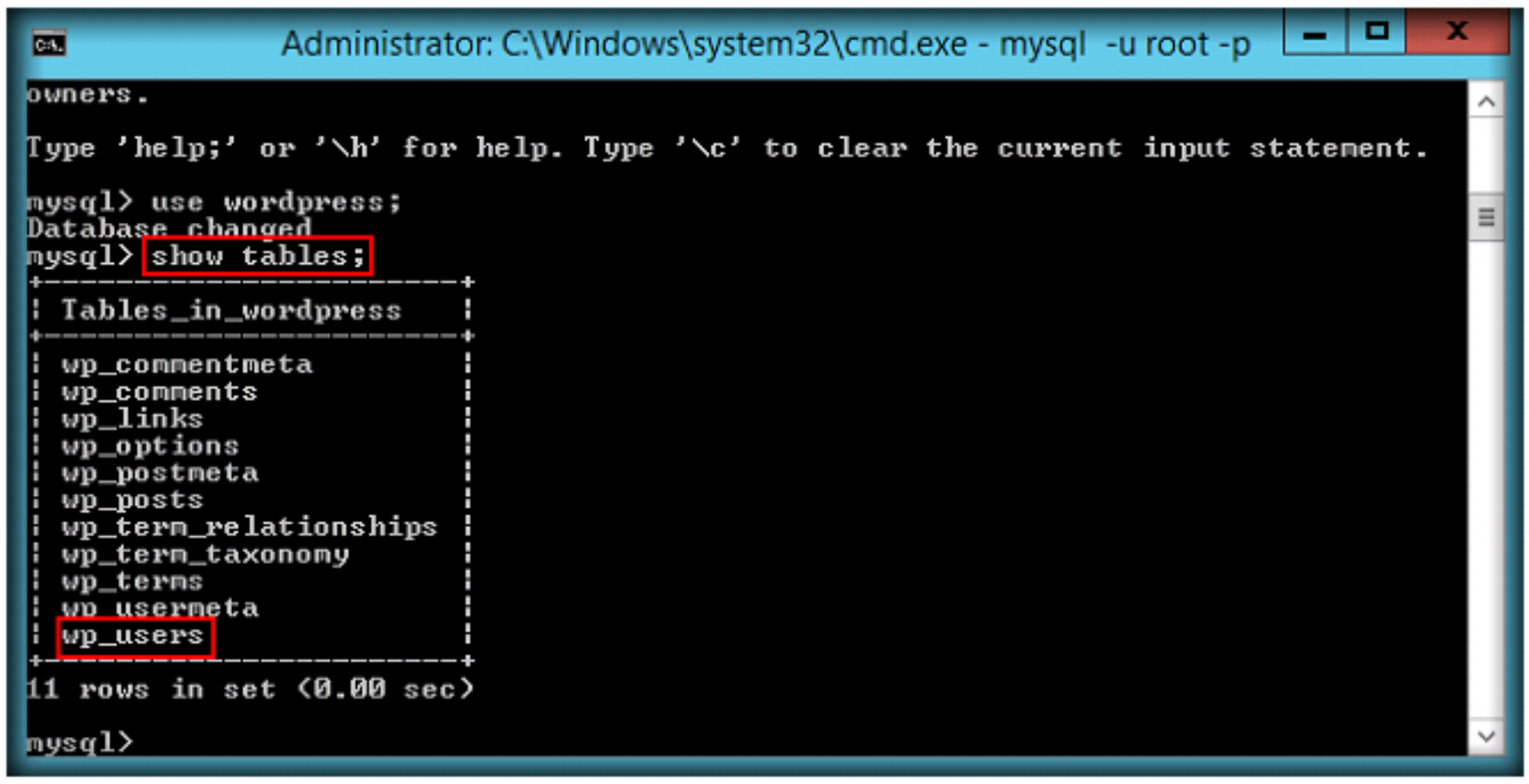

FIGURE 3.9: Viewing the tables in the database

16. The **wp_users** table contains all the users accounts associated with the wordpress website. To view the users, type **select * from wp_users;** and press **Enter**.

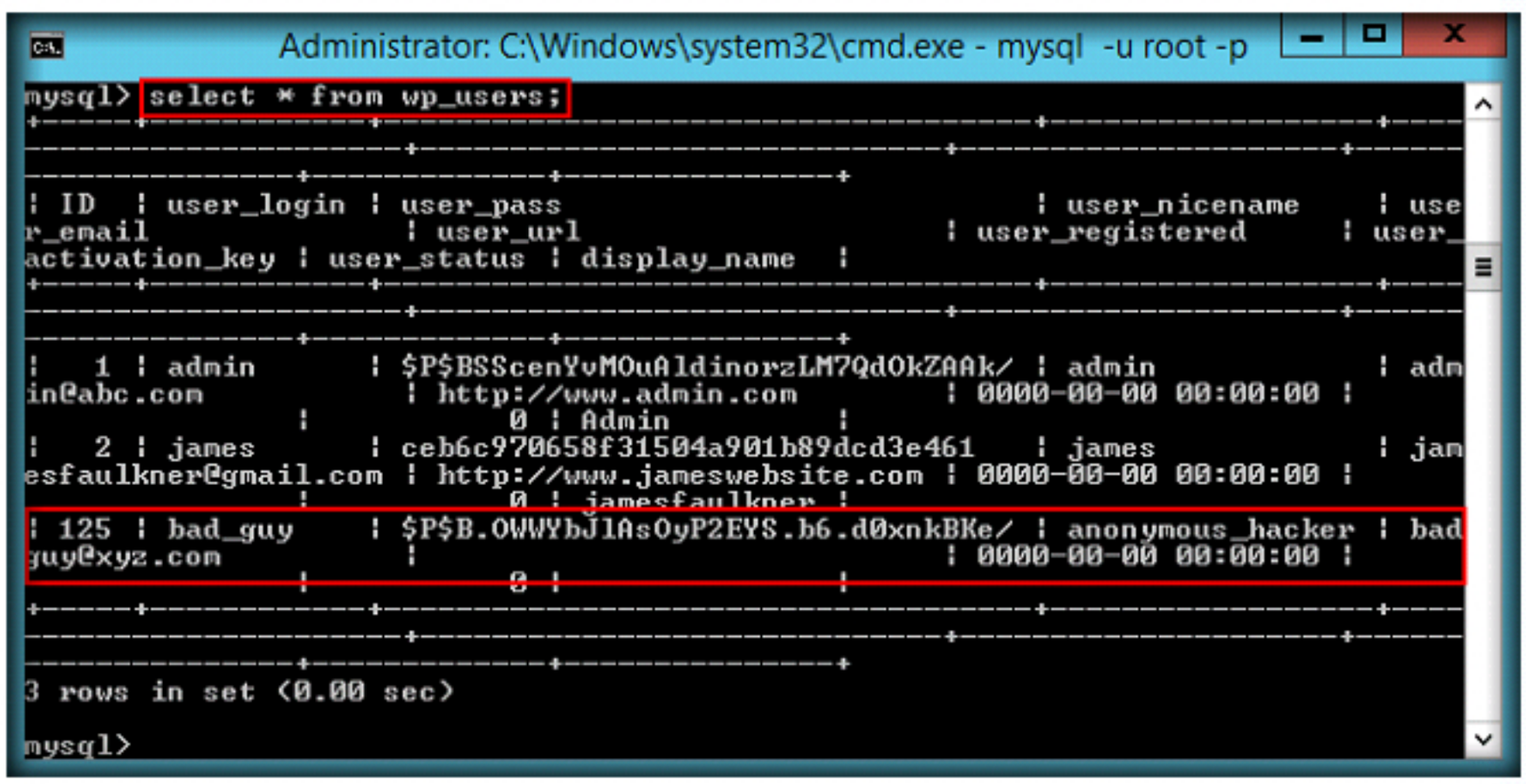

FIGURE 3.10: Viewing the contents of wp_users table

17. It is observed that a suspicious user account with the username **bad_guy** is present in the table. Make a note of the user ID which is **125**.

18. Since the scenario in the beginning of the lab states that a suspicious post was found on the webpage, we shall view the columns in **wp_posts** table. To view the columns, type **show columns in wp_posts;** and press **Enter**.

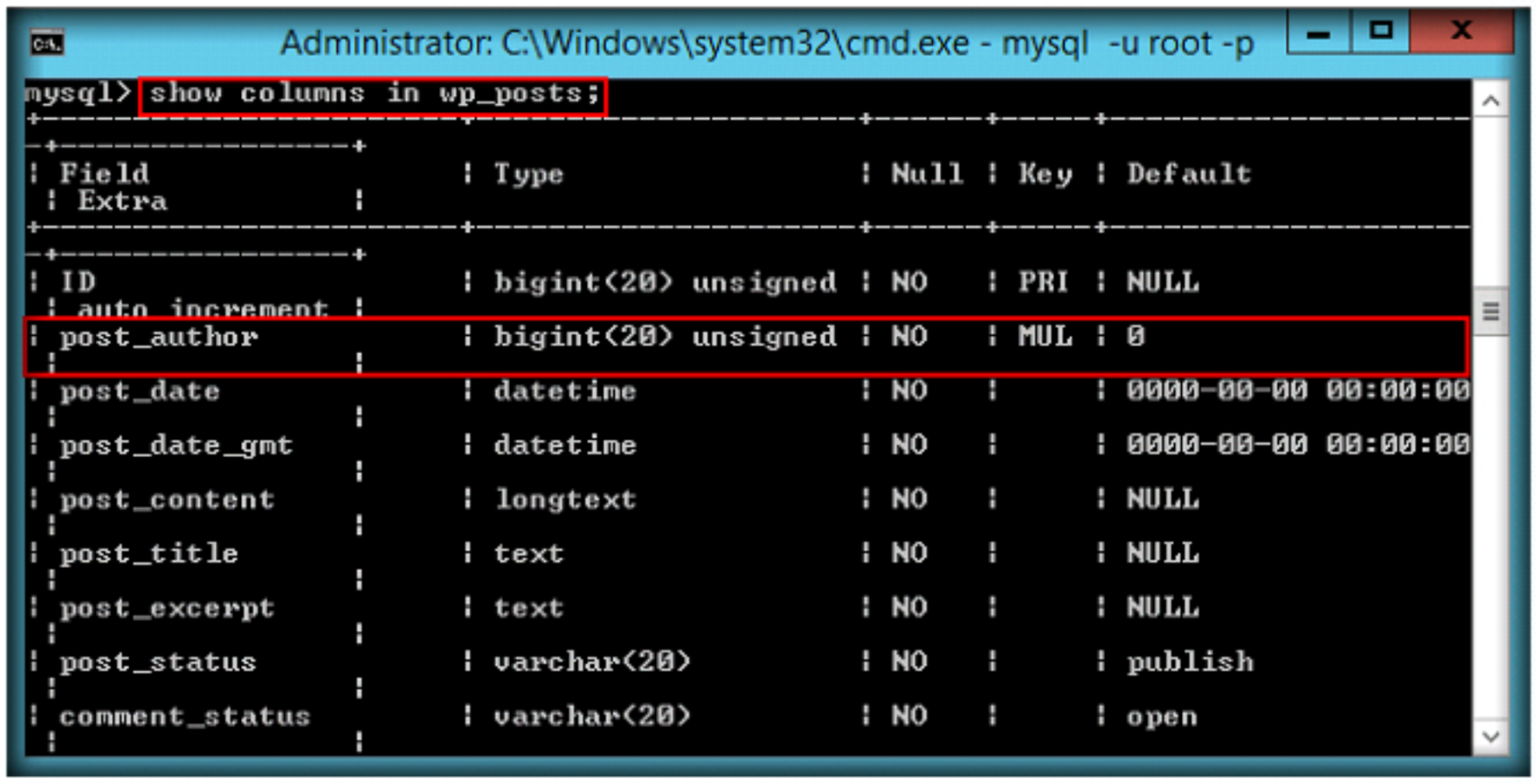

FIGURE 3.11: Viewing the columns in wp_posts table

19. You will observe a column named **post_author**, which corresponds to the posts made by the users.

20. Now, using **post_author** and the user id of **bad_guy**, we can collect all the posts made by the suspicious user (bad_guy).

21. Issue the following commands to collect the posts:

```
select * from wp_posts
where post_author = '125'
into outfile 'evidence.txt';
```

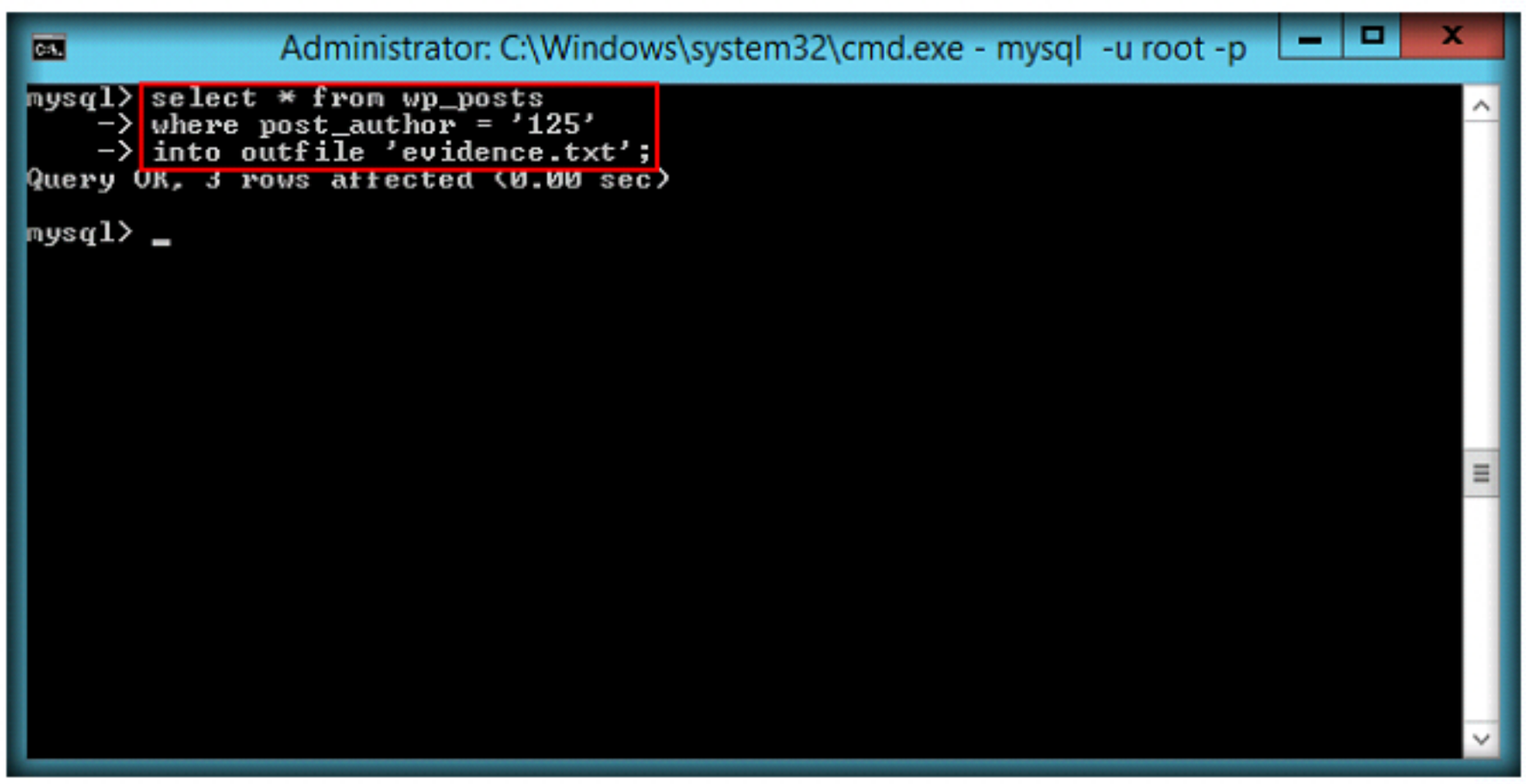

FIGURE 3.12: Issuing command in the mysql shell

22. By issuing the above commands, the posts made by the user whose ID is **125** are collected, and saved to a file named **evidence.txt** in the location **C:\wamp\bin\mysql\mysql5.6.17\data\wordpress**.

23. Now, navigate to this location and open the file. You will observe all the posts made by the user as shown in the following screenshot:

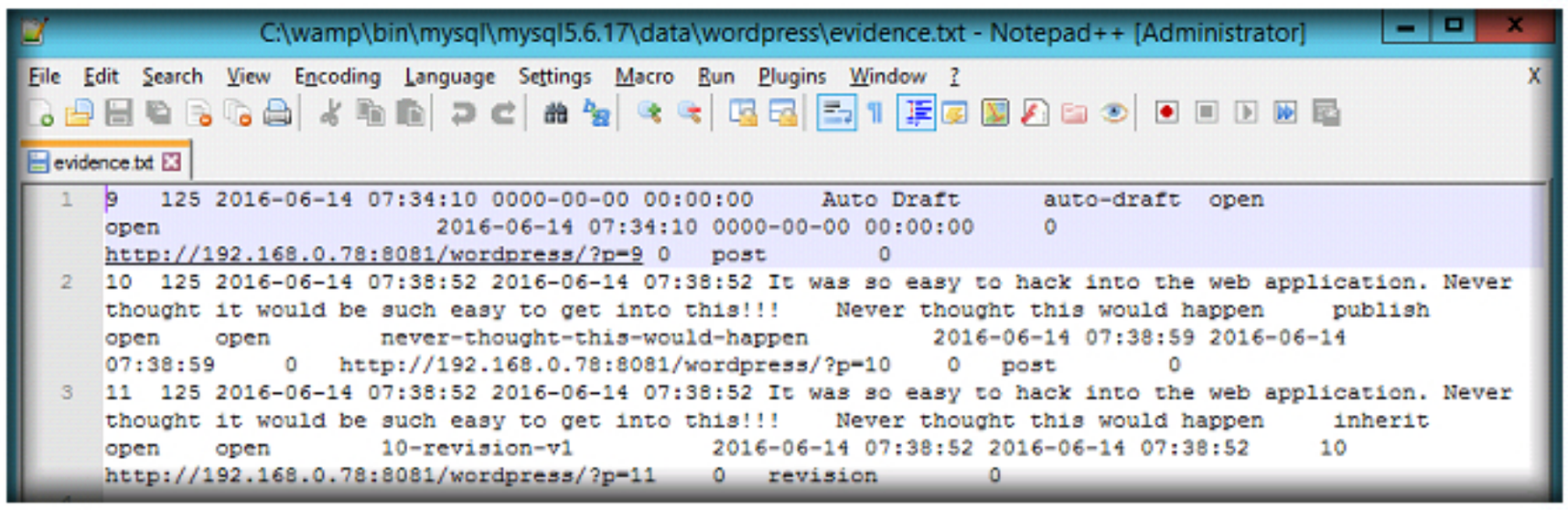

FIGURE 3.13: Viewing the contents of evidence.txt file

24. Now, we shall track events performed by the malicious user (MyISAM Storage Engine) and recover the deleted data.

25. The binary log files store all the transactions occurred on the databases. An investigator can examine these files to track the events performed by a particular user on the target database.

26. Navigate to **C:\CHFI-Tools\CHFIv9 Module 09 Database Forensics\Evidences\data**. You will find all the logs associated with the database as shown in the following screenshot:

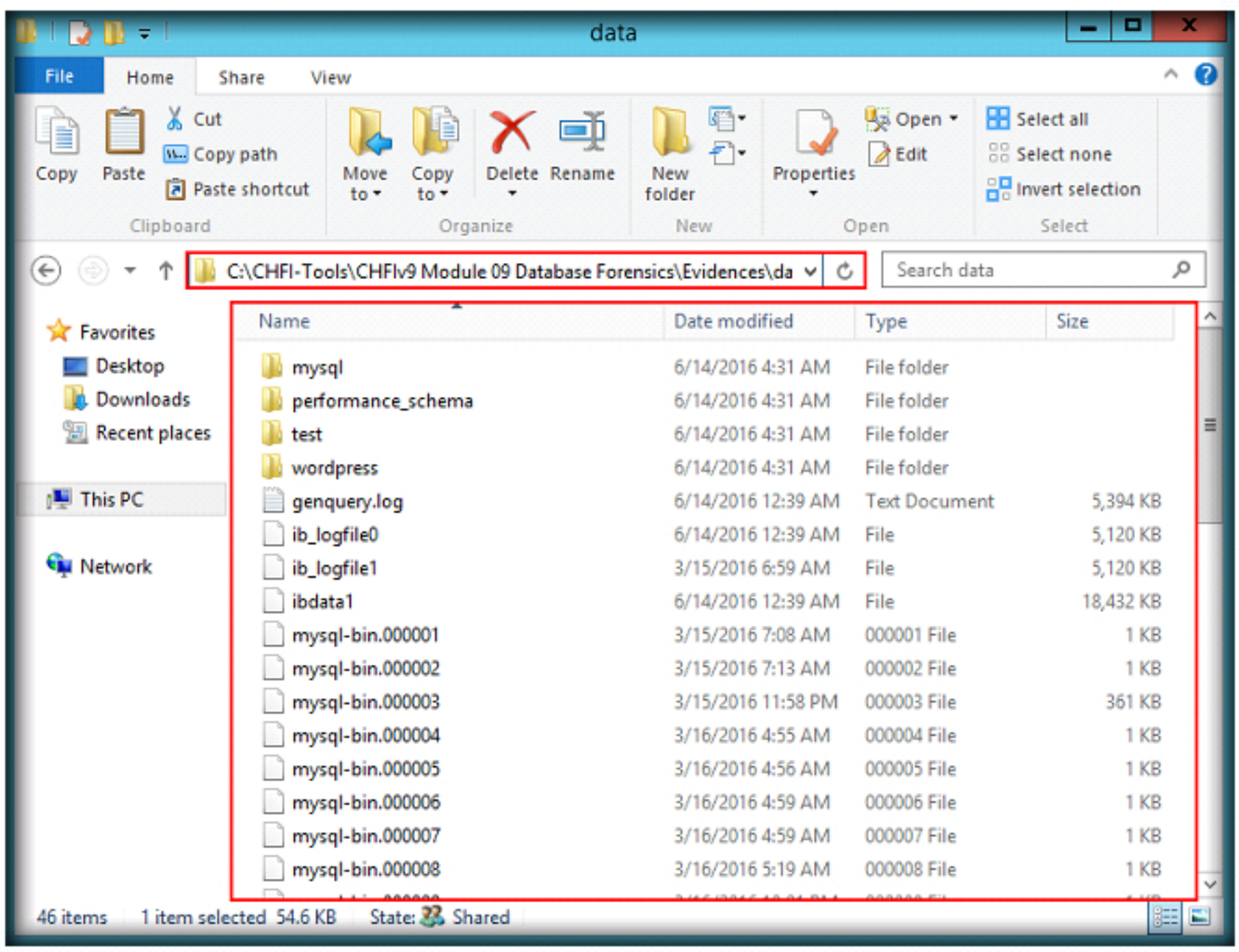

FIGURE 3.14: Examining log files

27. Analyzing the **.frm** files helps a forensic examiner to understand the table format and the terms related to the table content.
28. Since the malicious user created a user account for himself with the login name **bad_guy**, you may analyze the **wp_users.frm** file with a hex editor to view the column name (along with its hexa decimal equivalent) that contains a list of login names associated with the users.
29. Now, open **wordpress** folder, right-click **wp_users.frm** and select **Hex Edit with Hex Workshop v6.8** from the context menu.

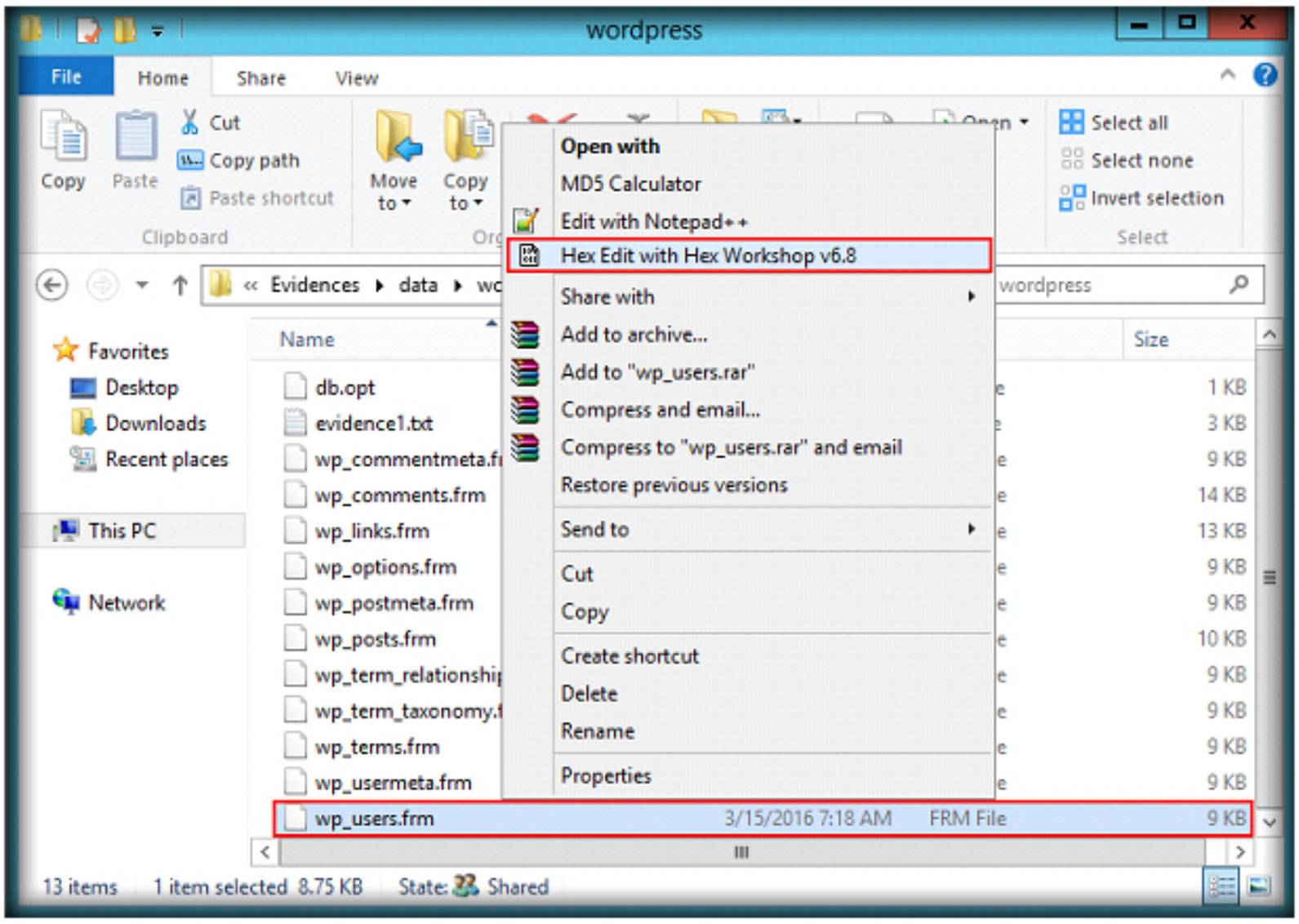

FIGURE 3.15: Viewing contents of wp_users.frm using hex editor

30. We can observe that the login names are stored under the **user_login** column whose hexadecimal equivalent is **757365725F6C6F67696E**.

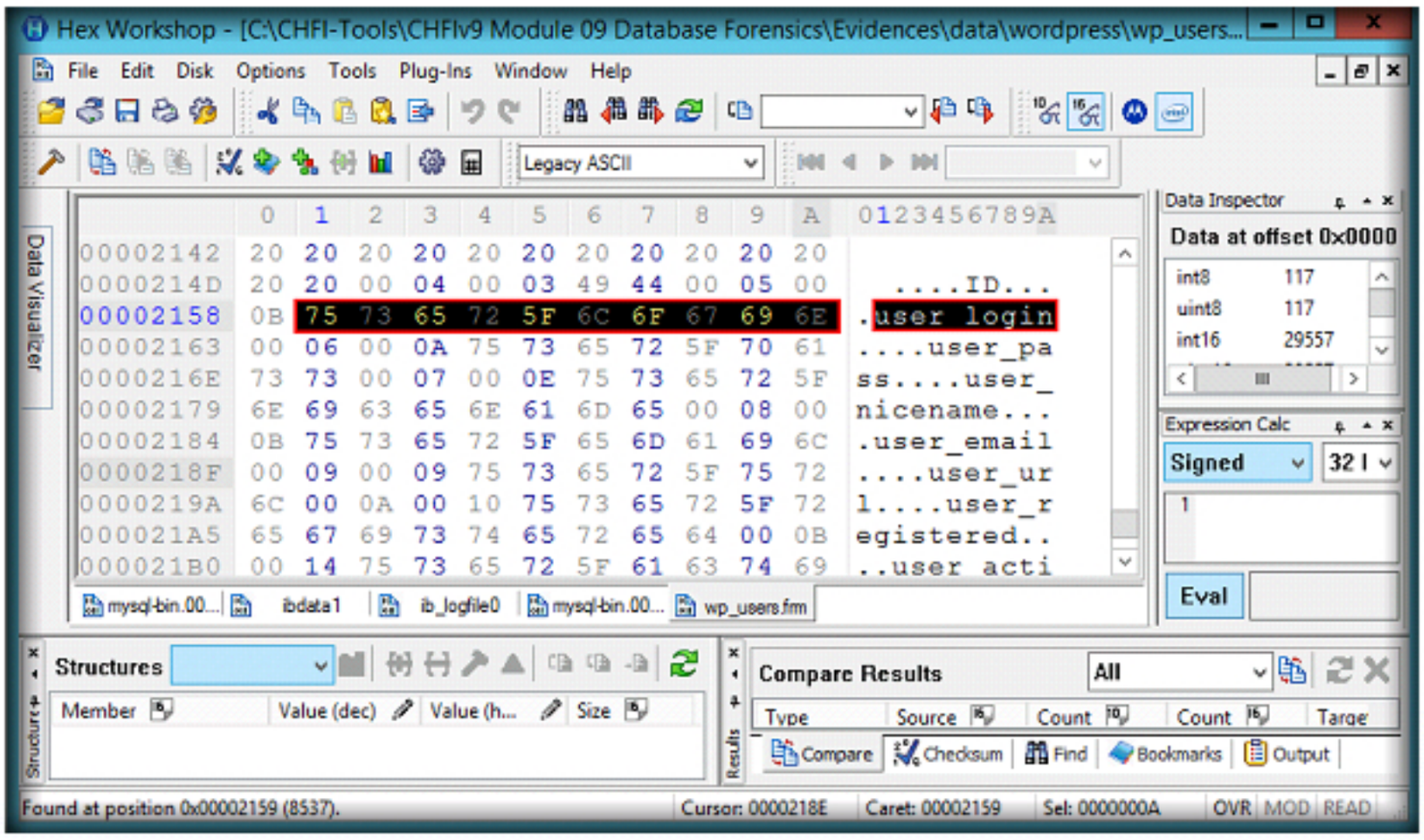

FIGURE 3.16: Viewing contents of wp_users.frm using hex editor

31. Using this phrase, we shall first find the attacker's login name, that is, **bad_guy** from the binary logs, and from there on, we shall trace the user activities performed by the malicious user.
32. In this lab, we shall analyze the **mysql-bin.000034** log file. Open the file with Hex Workshop.
33. Examine each binary log for the text string **user_login** or hex value **757365725F6C6F67696E**.

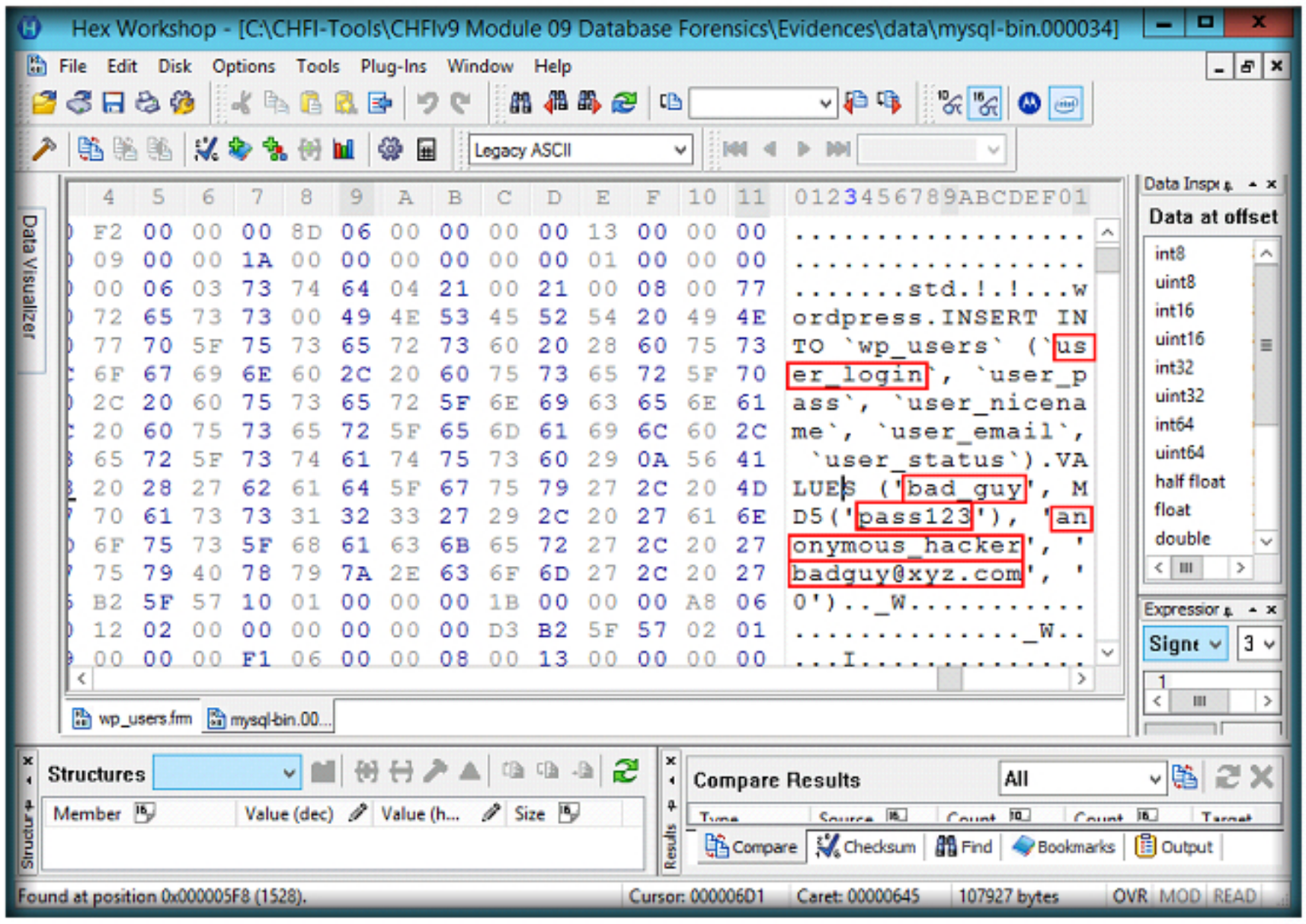

FIGURE 3.17: Viewing the contents of mysql-bin.000034 log file

34. While conducting a detailed examination on the binary files, we can find that one of the binary files recorded an event where a query is executed for creating a user account with the
    a. Login name – bad_guy
    b. Password – pass123
    c. Nice name – anonymous_hacker
    d. Email ID – badguy@xyz.com

35. In the same way, scroll down the binary logs one-by-one to see the logs corresponding to the malicious user's actions.

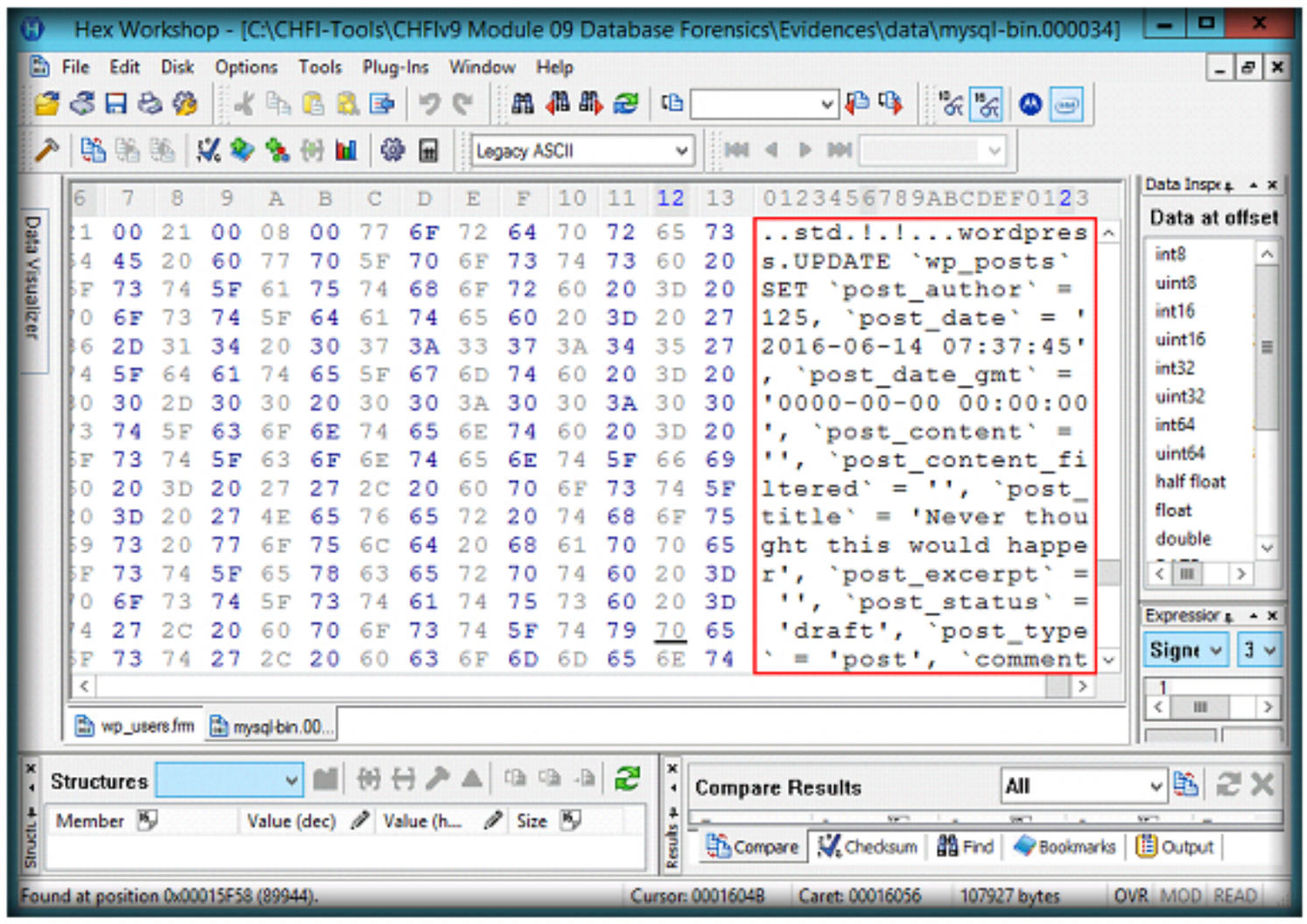

FIGURE 3.18: Viewing the contents of mysql-bin.000034 log file

36. We can observe that the attacker made a post (post_author id: 125) on 14th June, 2016 at GMT 07:37:45.

37. In the same way, you may search for all the actions performed by the attacker on the posts by looking for **`post_author` = 125** in the hex editor.

38. To find the actions performed by the attacker, press **Ctrl+F** on the keyboard. **Find** window appears. Select **Text String** from the **Type** drop-down list, enter **`post_author` = 125** in the **Value** text field, select **Down** radio-button under the **Direction** section and click **OK**.

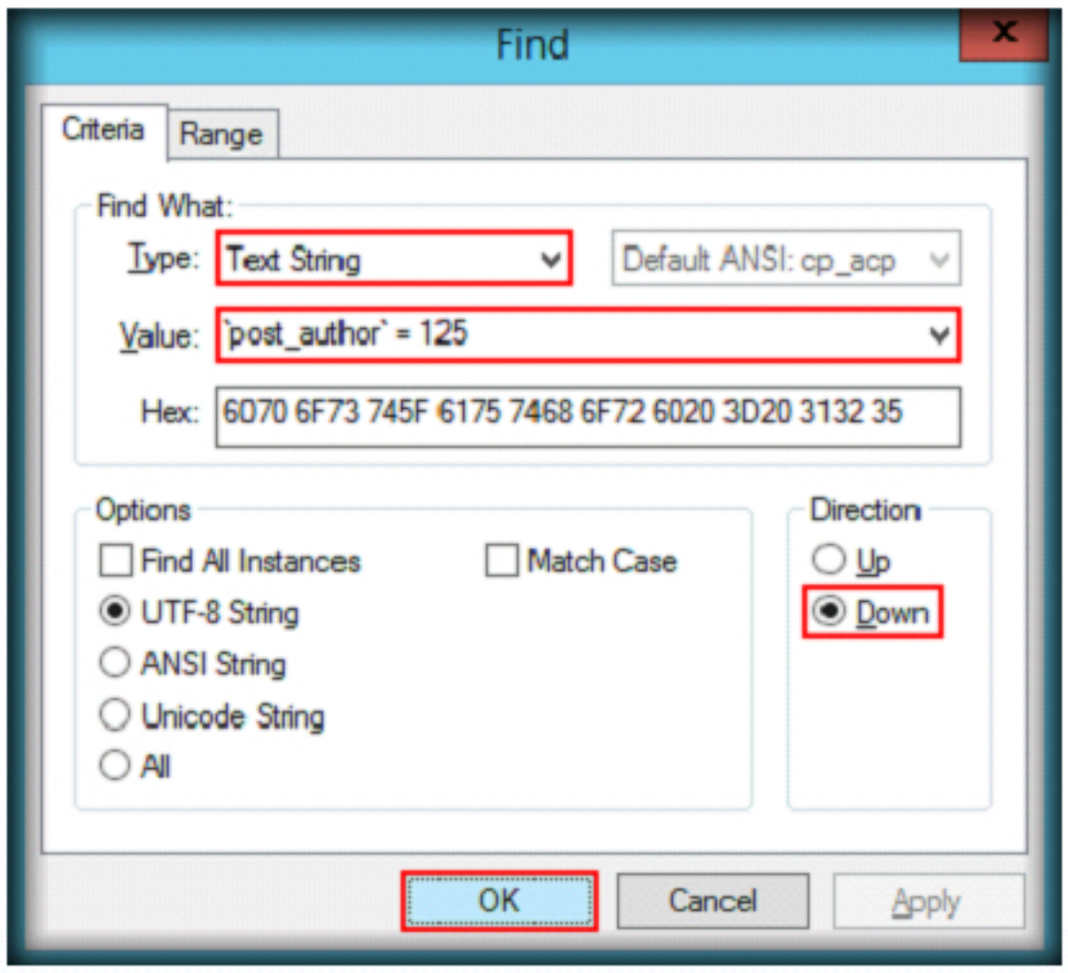

FIGURE 3.19: Find window

39. While you scroll down the log file **mysql-bin.000034**, you will come across various actions performed on the database like user account deletion, new posts on the website, etc.

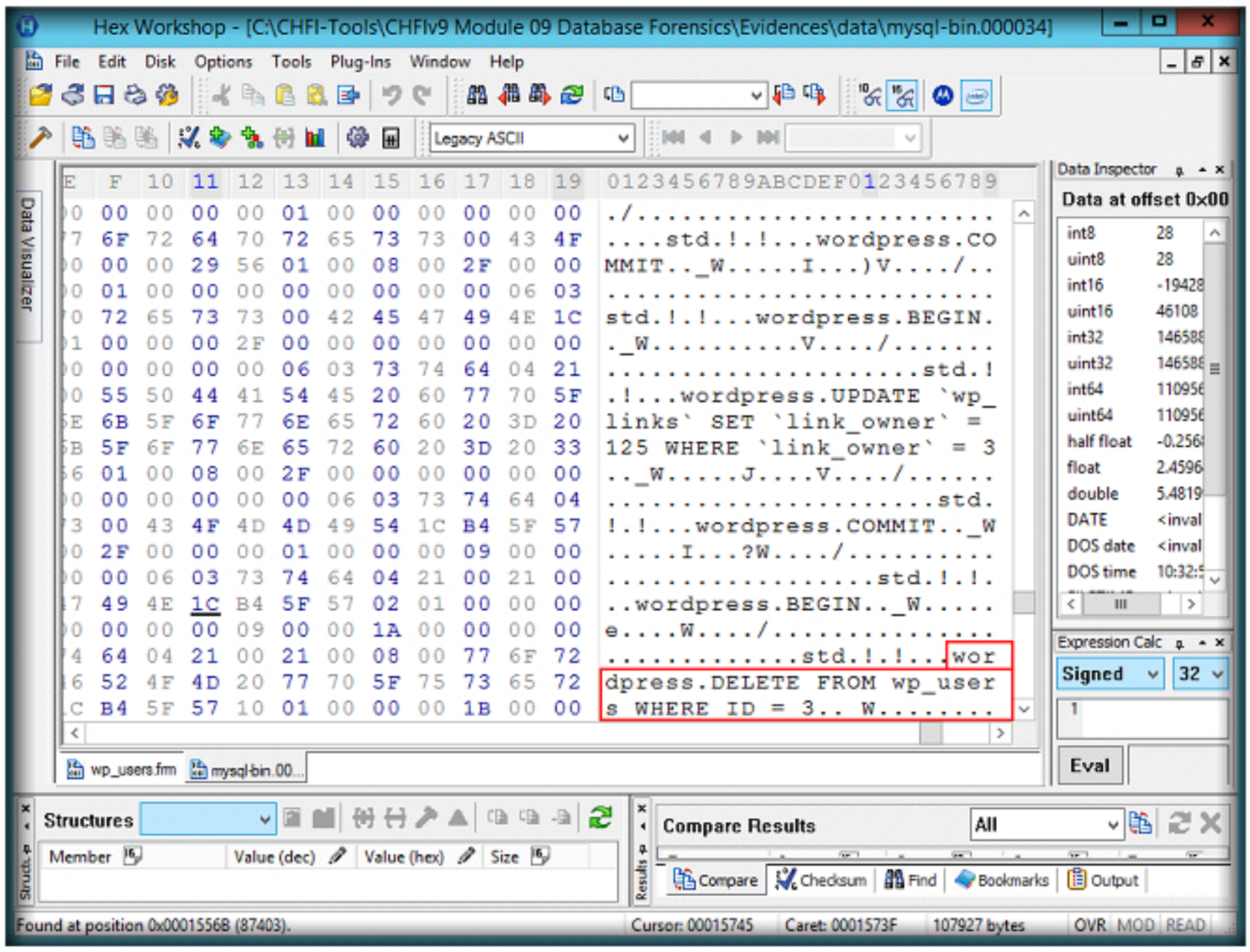

FIGURE 3.20: Viewing the contents of mysql-bin.000034 log file

40. In the above screenshot, you can observe a **MySQL** query for deleting a user associated with the user ID 3. In the same way, you may examine all the log files and find the transactions performed by the attacker.

Lab Analysis

Analyze the results and document the findings of the lab.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Internet Connection Required	
☐ Yes	☑ No
Platform Supported	
☑ Classroom	☑ iLabs