1.1 Core Assertions

Assertion	Req
SFT-CA-01. The MD5 (or SHA-1) hash value of the database, and associated journal mode file (e.g., -journal, -wal) shall not be altered between when analysis begins, and analysis is complete.	CR-01
SFT-CA-02. The associated journal mode file (e.g., -journal, -wal) shall not be deleted after analysis is complete.	CR-01
SFT-CA-03. The tool shall interpret the SQLite Page Size (in bytes).	CR-02
SFT-CA-04. The tool shall report the SQLite Journal Mode (write version)	CR-02
SFT-CA-05. The tool shall report the SQLite Journal Mode (read version)	CR-02
SFT-CA-06. The tool shall report the number of pages in the database	CR-02
SFT-CA-07. The tool shall report the SQLite database text encoding.	CR-02
SFT-CA-08. The tool shall report all table names for each table within the database.	CR-03
SFT-CA-09. The tool shall report all column names for each table in the database.	CR-03
SFT-CA-10. The tool shall report the number of rows for each table in the database.	CR-03
SFT-CA-11. The tool shall report on all recoverable rows that are contained within the database.	CR-04
SFT-CA-12. The tool shall report on all recoverable rows that are contained within the associated journal mode file (e.g., -journal, -wal).	CR-04
SFT-CA-13. The tool shall report the source file name for each recovered data element.	CR-05

1.2 Optional Test Assertions

Test Assertions for Optional Features	Req
SFT-AO-01. The tool shall report all CREATE TABLE statements for each table in the database.	RO-01
SFT-AO-02. The tool shall report the data type for each column within each table in the database.	RO-01
SFT-AO-03. The tool shall report which column is the primary key for each table in the database.	RO-01
SFT-AO-04. The tool shall report if the row was recovered because of a deletion or an update within the database.	RO-02
SFT-AO-05. The tool shall report if the row was recovered because of a deletion or an update in the associated journal mode file (e.g., -journal, -wal).	RO-02
SFT-AO-06. The tool shall report the file offset for each recovered data element presented.	RO-03
SFT-AO-07. The tool shall report the table name for each recovered data element presented.	RO-03
SFT-AO-08. The tool shall be able to present the sequence of transactions in the associated -wal file.	RO-04

2 SQLite Forensics Tool Test Cases

Core Test Cases	Assert
SFT-01. SQLite header parsing. This test case verifies that the tool provides the following (5) attributes as contained in the SQLite header: ▪ Page Size ▪ Journal Mode Information ▪ Number of Pages ▪ Text Encoding (i.e., UTF-8, UTF-16 Little Endian, and UTF-16 Big Endian) *Test Actions:* SFT-01-UTF8-WAL – Create SQLite file with specified parameters. 1. SQLITE3 SFT-01-UTF8-WAL.sqlite 2. PRAGMA journal_mode = WAL 3. PRAGMA encoding = 'UTF-8'; 4. PRAGMA page_size = 4096; 5. Create Table 6. Create 100 Rows of Data within Table 7. .quit 8. Read header and validate: Page Size, Journal Mode, Number of Pages, and Encoding. 9. If files have changed, investigate each set of test actions to determine where the change occured. *Test Actions:* SFT-01-UTF16BE-PERSIST – Create SQLite file with specified parameters. 1. SQLITE3 SFT-01-16BE-PERSIST.sqlite 2. PRAGMA journal_mode = PERSIST 3. PRAGMA encoding = 'UTF-16be'; 4. PRAGMA page_size = 1024; 5. Create Table 6. Create 100 Rows of Data within Table 7. .quit 8. Read header and validate: Page Size, Journal Mode, Number of Pages, and Encoding. 9. If files have changed, investigate each set of test actions to determine where the change occured. *Test Actions:* SFT-01-UTF16LE-OFF – Create SQLite file with specified parameters. 1. SQLITE3 SFT-01-16LE-OFF.sqlite 2. PRAGMA journal_mode = OFF 3. PRAGMA encoding = 'UTF-16le'; 4. PRAGMA page_size = 8192; 5. Create Table	CA-01 CA-03 CA-04 CA-05 CA-06 CA-07

6. Create 100 Rows of Data within Table 7. .quit 8. Read header and validate: Page Size, Journal Mode, Number of Pages, and Encoding. 9. If files have changed, investigate each set of test actions to determine where the change occured. *Conformance Indicator:* Data reported matches data contained within header as specified in each test action.	
SFT-02. SQLite Schema Reporting. This test case verifies that the tool provides a listing of all: 1. Tables 2. Column names for each table 3. Row information for each table *Test Actions:* SFT-02 – Schema Reporting 1. SQLITE3 SFT-02.sqlite 2. Create Table with at least (5) columns 3. Create 100 Rows of Data within Table 4. Read table data and validate: Table Names, Column Names, and number of rows. 5. If files have changed, investigate each set of test actions to determine where the change occured. *Conformance Indicator:* Data reported matches data contained within the database as specified in each test action.	CA-01 CA-08 CA-09 CA-10
SFT-03. SQLite Recoverable Rows This test case verifies that the tool reports the file name (e.g., source) and recovered information for all recoverable rows (e.g., deleted and updated): ▪ SQLite database file ▪ SQLite database journal mode file (e.g., -journal, -wal) *Test Actions:* SFT-03-PERSIST – Create SQLite file with an associated -journal file. 1. SQLITE3 SFT-03-PERSIST.sqlite 2. PRAGMA journal_mode = PERSIST 3. Create Table 4. Create 10,000 rows of data within table 5. Delete 100 rows (randomly) 6. Modify 100 rows (randomly) 7. Hard Stop (e.g., CTRL + C) 8. Perform SQLite database recovery 9. Validate reporting of 100 deleted rows and 100 modified rows.	CA-01 CA-02 CA-11 CA-12

10. If files have changed, investigate each set of test actions to determine where the change occured. *Test Actions:* SFT-03-WAL – Create SQLite file with an associated -wal file. 1. SQLITE3 SFT-03-WAL.sqlite 2. PRAGMA journal_mode = WAL 3. Create Table 4. Create 10,000 rows of data within table 5. Delete 100 rows (randomly) 6. Modify 100 rows (randomly) 7. Hard Stop (e.g., CTRL + C) 8. Perform SQLite database recovery 9. Validate reporting of 100 deleted rows and 100 modified rows. 10. If files have changed, investigate each set of test actions to determine where the change occured. *Conformance Indicator:* Data reports deleted, and modified row data as specified in each test action.	
SFT-04. SQLite Data Element metadata This test case verifies that the tool reports the file name (e.g., source) for all recovered data elements: ▪ SQLite database file ▪ SQLite database journal mode file (e.g., -journal, -wal) *Test Actions:* SFT-04 – PERSIST 1. Using SQLITE3 SFT-03-PERSIST.sqlite and SQLITE3 SFT-03-PERSIST.sqlite-journal 2. Perform SQLite database recovery 3. Verify that tool reports the file name (e.g., source) where each recoverable data element is located. 4. If files have changed, investigate each set of test actions to determine where the change occured. *Test Actions:* SFT-04 – WAL 1. Using SQLITE3 SFT-03-WAL.sqlite and SQLITE3 SFT-03-WAL.sqlite-wal 2. Perform SQLite database recovery 3. Verify that tool reports the file name (e.g., source) where each recoverable data element is located. 4. If files have changed, investigate each set of test actions to determine where the change occured. *Conformance Indicator:* Data reports deleted, and modified row data as specified in each test action.	CA-01 CA-02 CA-13

Optional Test Cases	
SFT-05. SQLite schema data reporting This test case verifies that the tool reports the SQLite metadata for, all create table statements, type (e.g., Storage Class, datatype, or affinity) for each column, and identify which column is the primary key for each table in the database. *Test Actions:* SFT-05 – Schema Reporting 1. SQLITE3 SFT-05.sqlite 2. Create Table with at least (5) columns: Primary Key, Int, Float, Text, Blob, Boolean 3. Create 100 Rows of Data within Table 4. Read table data and report all create table statements, associated column types and the primary key for each table. 5. If files have changed, investigate each set of test actions to determine where the change occured. *Conformance Indicator:* Data reported matches data contained within the database as specified in each test action.	CA-01 AO-01 AO-02 AO-03
SFT-06. Recovered row metadata This test case verifies that the tool reports the recovered row because of either a deletion or an update within the database file, or the associated journal mode file (e.g., -journal, -wal). *Test Actions:* SFT-06 – PERSIST 1. Using SQLITE3 SFT-03-PERSIST.sqlite and SQLITE3 SFT-03-PERSIST.sqlite-journal 2. Perform SQLite database recovery 3. Tool reports the file name (e.g., source) and if the row was the result of an update or a deletion. 4. If files have changed, investigate each set of test actions to determine where the change occured. *Test Actions:* SFT-06 – WAL 1. Using SQLITE3 SFT-03-WAL.sqlite and SQLITE3 SFT-03-WAL.sqlite-wal 2. Perform SQLite database recovery 3. Tool reports the file name (e.g., source) and if the row was the result of an update or a deletion. 4. If files have changed, investigate each set of test actions to determine where the change occured. *Conformance Indicator:* Recovered information matches the actions from each test case.	CA-01 CA-02 AO-04 AO-05
SFT-07. SQLite recovered data information	CA-01

This test case verifies that the tool reports the following metadata for all recoverable data elements: 1. Offset within the file 2. Identify the table name associated with the row *Test Actions:* SFT-07 – PERSIST 1. Using SQLITE3 SFT-03-PERSIST.sqlite and SQLITE3 SFT-03-PERSIST.sqlite-journal 2. Perform SQLite database recovery 3. Tool reports the offset and length of the data within the payload for each recovered cell. 4. Tool reports the table name for each row of recovered data. 5. If files have changed, investigate each set of test actions to determine where the change occured. *Test Actions:* SFT-07 – WAL 1. Using SQLITE3 SFT-03-WAL.sqlite and SQLITE3 SFT-03-WAL.sqlite-wal 2. Perform SQLite database recovery 3. Tool reports the offset and length of the data within the payload for each recovered cell. 4. Tool reports the table name for each row of recovered data. 5. If files have changed, investigate each set of test actions to determine where the change occured. *Conformance Indicator:* Recovered metadata matches the actions from each test case.	CA-02 AO-06 AO-07
SFT-08. Journal sequencing/wal timelining This test case verifies that the tool reports the sequence of transactions in the associated -wal file. *Test Actions:* SFT-08 – WAL 1. Using SQLITE3 SFT-03-WAL.sqlite and SQLITE3 SFT-03-WAL.sqlite-wal 2. Perform SQLite database recovery 3. Order recovered transactions within the -wal journal file. 4. If files have changed, investigate each set of test actions to determine where the change occured. *Conformance Indicator:* Recovered data is sequenced matching the chronological actions executed during testing (SFT-08-WAL).	CA-01 CA-02 AO-08