SQLite3 Presentation

Function: sqlite3_close()

Submitted by:

Name : Mosamma Sultana Trina
Roll : BSSE 1313
Course: CSE 404

sqlite3_close()

The sqlite3_close() function is part of the SQLite library, which is a C library that provides a lightweight, self-contained, serverless, and zero-configuration SQL database engine. The function sqlite3_close() is used to close an SQLite database connection that was previously opened using the sqlite3_open() or related functions.

```
int sqlite3_close(sqlite3*);
```

When we call sqlite3_close(), SQLite performs several important tasks:

- **It finalizes any pending prepared statements or compiled queries associated with the database connection.**
- **It releases any memory and resources held by the database connection object.**
- **It ensures that any changes made to the database are properly written to disk.**
- **It's important to note that if you close a database connection without finalizing prepared statements or completing transactions, you might encounter resource leaks or data integrity issues.**

Here's the sequence of function calls and operations performed by the sqlite3Close function:

1. Check if the db pointer is NULL.
   - If db is NULL, return SQLITE_OK.
2. Check if the database (db) is in a healthy state by calling sqlite3SafetyCheckSickOrOk.
   - If the database is not in a healthy state, return SQLITE_MISUSE_BKPT.
3. Enter the database mutex (db->mutex) for synchronization.
   - Check if tracing is enabled for SQLITE_TRACE_CLOSE and call the associated trace callback.
4. Call disconnectAllVtab to force the xDisconnect method on all virtual tables associated with the database.

5. Call sqlite3VtabRollback to handle open transactions and ensure that xDisconnect is called on virtual tables.

6. Check if the forceZombie flag is not set and if there are active SQL statements still in progress (database is busy).

- If true, return SQLITE_BUSY with an error message.

7. Check if SQLITE_ENABLE_SQLLOG is defined and call the SQL logging callback if registered.

8. Set the db->eOpenState to SQLITE_STATE_ZOMBIE.

9. Call sqlite3LeaveMutexAndCloseZombie to release the mutex, perform additional cleanup, and close the database.

10. Call sqlite3LeaveMutexAndCloseZombie to release the mutex, perform additional cleanup, and close the database.

11. Finally, return SQLITE_OK to indicate successful closure.


THANK
YOU