Q - A. WHAT IS THE USE OF SQLITE OPEN HELPER CLASS INSQLITE?

Ans.

A helper class to manage database creation and version management.

You create a subclass implementing [onCreate(SQLiteDatabase)](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html" \l "onCreate(android.database.sqlite.SQLiteDatabase)), [onUpgrade(SQLiteDatabase, int, int)](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html" \l "onUpgrade(android.database.sqlite.SQLiteDatabase, int, int)) and optionally [onOpen(SQLiteDatabase)](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html" \l "onOpen(android.database.sqlite.SQLiteDatabase)), and this class takes care of opening the database if it exists, creating it if it does not, and upgrading it as necessary. Transactions are used to make sure the database is always in a sensible state.

This class makes it easy for [ContentProvider](https://developer.android.com/reference/android/content/ContentProvider.html) implementations to defer opening and upgrading the database until first use, to avoid blocking application startup with long-running database upgrades.

|  |  |
| --- | --- |
| **Public constructors** | |
| [SQLiteOpenHelper](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#SQLiteOpenHelper(android.content.Context, java.lang.String, android.database.sqlite.SQLiteDatabase.CursorFactory, int))([Context](https://developer.android.com/reference/android/content/Context.html) context, [String](https://developer.android.com/reference/java/lang/String.html) name, [SQLiteDatabase.CursorFactory](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.CursorFactory.html) factory, int version)  Create a helper object to create, open, and/or manage a database. |  |
| [SQLiteOpenHelper](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#SQLiteOpenHelper(android.content.Context, java.lang.String, android.database.sqlite.SQLiteDatabase.CursorFactory, int, android.database.DatabaseErrorHandler))([Context](https://developer.android.com/reference/android/content/Context.html) context, [String](https://developer.android.com/reference/java/lang/String.html) name, [SQLiteDatabase.CursorFactory](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.CursorFactory.html) factory, int version, [DatabaseErrorHandler](https://developer.android.com/reference/android/database/DatabaseErrorHandler.html)errorHandler)  Create a helper object to create, open, and/or manage a database. |  |

|  |  |
| --- | --- |
| **Public methods** | |
| void | [close](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#close())()  Close any open database object. |
| [String](https://developer.android.com/reference/java/lang/String.html) | [getDatabaseName](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#getDatabaseName())()  Return the name of the SQLite database being opened, as given to the constructor. |
| SQLiteDatabase | [getReadableDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#getReadableDatabase())()  Create and/or open a database. |
| [SQLiteDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) | [getWritableDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#getWritableDatabase())()  Create and/or open a database that will be used for reading and writing. |
| void | [onConfigure](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#onConfigure(android.database.sqlite.SQLiteDatabase))([SQLiteDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) db)  Called when the database connection is being configured, to enable features such as write-ahead logging or foreign key support. |
| abstract void | [onCreate](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#onCreate(android.database.sqlite.SQLiteDatabase))([SQLiteDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) db)  Called when the database is created for the first time. |
| void | [onDowngrade](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#onDowngrade(android.database.sqlite.SQLiteDatabase, int, int))([SQLiteDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) db, int oldVersion, int newVersion)  Called when the database needs to be downgraded. |
| void | [onOpen](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#onOpen(android.database.sqlite.SQLiteDatabase))([SQLiteDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) db)  Called when the database has been opened. |
| abstract void | [onUpgrade](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#onUpgrade(android.database.sqlite.SQLiteDatabase, int, int))([SQLiteDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) db, int oldVersion, int newVersion)  Called when the database needs to be upgraded. |
| void | [setWriteAheadLoggingEnabled](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html#setWriteAheadLoggingEnabled(boolean))(boolean enabled)  Enables or disables the use of write-ahead logging for the database. |

|  |
| --- |
| **Inherited methods** |
| [https://developer.android.com/assets/images/styles/disclosure_down.png](https://developer.android.com/reference/android/database/sqlite/SQLiteOpenHelper.html)From class [java.lang.Object](https://developer.android.com/reference/java/lang/Object.html) |

B. WHAT IS THE USE OF ONUPGRADE FUNCTION IN SQLITEOPENHELPER CLASS?

Ans.

Syntax.

void onUpgrade ([SQLiteDatabase](https://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html) db,

int oldVersion,

int newVersion)

|  |  |
| --- | --- |
| Parameters | |
| db | SQLiteDatabase: The database. |
| oldVersion | int: The old database version. |
| newVersion | int: The new database version. |

Usage:

Called when the database needs to be upgraded. The implementation should use this method to drop tables, add tables, or do anything else it needs to upgrade to the new schema version.

The SQLite ALTER TABLE documentation can be found [here](http://sqlite.org/lang_altertable.html). If you add new columns you can use ALTER TABLE to insert them into a live table. If you rename or remove columns you can use ALTER TABLE to rename the old table, then create the new table and then populate the new table with the contents of the old table.

This method executes within a transaction. If an exception is thrown, all changes will automatically be rolled back.

C. HOW TO SHOW SQLITE DATABASE TABLE INFORMATION IN ANDROID APPLICATION WHAT IS THE BEST WAY TO DO IT?

Ans.

Table Layout Using Cursor.

Showing data base information will be better suited with table layout. since table layout is not an adapter view, you can't use cursor adapter with it. So use table layout with cursor to show data base table information.

Cursors store query result records in rows and grant many methods to access and iterate through the records. Cursors should be closed when no longer used, and will be deactivated with a call to Cursor.deactivate() when the application pauses or exists. On resume the Cursor.requery() statement is executed to re-enable the Cursor with fresh data. These functions can be managed by the parent Activity by calling startManagingCursor().