**SQLITE** àCRUD(CREATE,READ,UPDATE,DELETE)

* SQLite database is an opensource, lightweight, no network access, and standalone SQL database that stores data to a text file on a device.
* SQLite database supports all the relational database features.
* SQLite is a build-in database implementation that comes with Android. You don’t need to establish any connections like JDBC, ODBC, or any other external connection for it, like what you will need to do in java applications.
* In Android, whenever an application needs to store a large amount of data the SQLite is the most preferred.
* It is better than any other repository systems like **SharedPreferences** or saving data in files. For many applications in Android, SQLite is the backbone of the app, whether it’s used directly or via some third-party wrapper.

SQLite Database Package :

* To use the SQLite database in the Android application, the main package will be android.database.sqlite.

**Three main components are used to handle SQLite operations in Android:**

* SQLiteOpenHelper Class The SQLiteOpenHelper class is used to determine the **name and version of the database** used in this class. It is responsible for opening the database if it exists, creating if it does not exists and upgrading if required.
* There are the onCreate() and onUpgrade() methods.

onCreate() invoked if the database doesn’t exist, meaning the very first time dealing with the SQLite database. Moreover, the onUpgrade() method is used to update the database schema to the most recent or, let’s say, the existing without losing the data.

**SQLiteDatabase in Android**

* The base class of the SQLite database class is SQLiteDatabase.
* Android has its implementation to perform CRUD (Create, Read, Update, Delete) operations.
* The SQLiteDatabase have methods such as insert(), update(), delete() and also execSQL() which are used for the execution of the database.

**Cursor**

* The query function will return the cursor object, which results from a query. One record is equal to the row of the query results.
* In Android, the cursor can perform buffer query results efficiently as it does not need to load the data into the memory.
* The most commonly used methods of the cursor are getCount(), moveToFirst(), moveToNext(). The getCount() used to get the number of records from the query results. The moveToFirst() used to move to the first record of the line. And the moveToNext() used to move to the next line.
* The query can be built in the SQLite database with the methods such as rawQuery() and query() using the SQLiteQueryBuilder class. The rawQuery() method accepts an SQL statement as input.
* The query() method has a structured interface in terms of parameters to specify the SQL query. The SQLiteQueryBuilder class allows you to create SQL queries.

**Example of Query()**

* return database.query(EmployeeTable,new String[]{key\_1,key\_2,...},null,null,null,null,null); 1
* return database.query(EmployeeTable,new String[]{key\_1,key\_2,...},null,null,null,null,null);

**Example of rawQuery()**

Cursor cursor = getReadableDatabase();

rawQuery("select \* from EmployeeTable);

1

2

Cursor cursor = getReadableDatabase();

rawQuery("select \* from EmployeeTable);