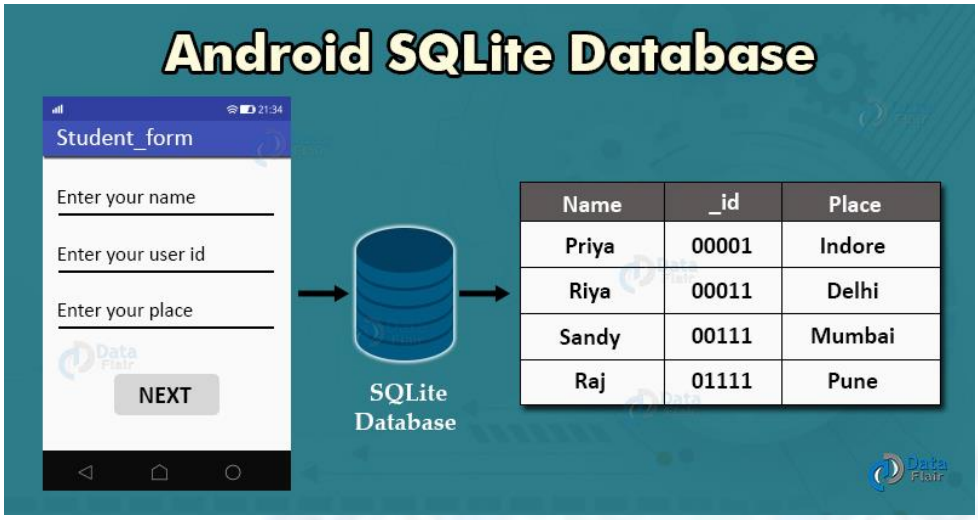
MANAGING DATA USING SQLITE,



What is SQLite?

SQLite is basically a Relational Database Management System (RDBMS), same as SQL. It is an open-source in-process library that is self-contained, serverless, has zero-configuration, and a transactional SQL database engine. Here, zero-configuration means unlike other database management systems, it doesn't need to be configured on the devices. The Lite here in SQLite is in terms of its setup, the database administration, and all the required resources.

Features of Android SQLite

SQLite has many features which it supports as follows:

- Full-featured SQL implementation along with various advanced capabilities such as partial indexing, JSON, and some other.
- Very simple with easy to use API.
- Has a really fast execution at times, and it is even faster than the direct file system Input-Output.
- Self-dependent as it has no external dependencies.
- Transactions in SQlite also have the ACID property that is – Atomicity, Consistency, Isolation, and Durability.
- Cross-platform as it supports Android, iOS, Linux, Mac, Windows, VxWorks, Solaris. It is easy to port it to other systems.
- Comes along with a standalone command-line interface client.

- Completely stored in a single cross-platform.
- Written completely in ANSI-C.
- Highly lightweight and small, as it is possible to configure it in less than 400Kbs. If configured by omitting the optional features, it uses less than 250Kbs.
- Custom functions are possible to implement.
- Allows the use of virtual tables
- Allows the use of procedures and triggers in SQL
- Free to use without any payment and subscription.

SQLite in Android

So, we know that SQLite is an open-source RDBMS used to perform operations on the databases stored in the form of rows and columns.

SQLite is highly supported by Android; in fact, Android comes with the built-in database implementation of SQLite. It is available on each and every Android database and emphasizes scalability, centralization, concurrency, and control. This strives to provide storage at the local level for applications and devices. It is highly economical, efficient, reliable, and independent. It is very simple and doesn't need to be compared with client/server databases.

1. SQLite supports the following three types of data:

- **Text Type** – to store strings or character type data.
- **Integer** Type – to store the integer data type.
- **Real Type** – to store long values.

In SQLite, the data types that are used are termed as valid; it is not validated by SQLite.

2. To use SQLite in Android applications, we use the package **android.database.sqlite**. This package contains all the APIs to use SQLite.

3. **SQLiteOpenHelper** is a class that is useful to create the database and manage it. The two constructors of SQLiteOpenHelper class are:

- **SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version_no)** – It creates objects for creating, opening, and managing the database.

- **SQLiteOpenHelper(Context context, String name, SQLiteDatabase.CursorFactory factory, int version_no, DatabaseErrorHandler errorHandler**) – Its object specifies the error handler along with creating, opening, and managing the database.

4. **SQLiteDatabase** is a class that has methods to perform operations such as create, update, delete, etc.
There are many methods that it includes, and a few of them are as follows :

Methods	Description
Void execSQL(String sql_query)	It will execute SQLquery.
Long insert(String table_name, String nullColumnHack, ContentValues values)	It will insert the record in the table with the values that are passed. The second argument makes sure that no null value is allowed._name
Int update(String table_name, ContentValues values, String where_clause, String[] where_args)	It will update the a that satisfies the where clause.

Creating And Updating Database In Android

For creating, updating and other operations you need to create a subclass or *SQLiteOpenHelper* class. SQLiteOpenHelper is a helper class to manage database creation and version management. It provides two methods onCreate(SQLiteDatabase db), onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion).

The SQLiteOpenHelper is responsible for opening database if exist, creating database if it does not exists and upgrading if required. The SQLiteOpenHelper only require the DATABASE_NAME to create database. After extending SQLiteOpenHelper you will need to implement its methods onCreate, onUpgrade and constructor.

onCreate(SQLiteDatabase sqLiteDatabase) method is called only once throughout the application lifecycle. It will be called whenever there is a first call to getReadableDatabase() or getWritableDatabase() function available in super SQLiteOpenHelper class. So SQLiteOpenHelper class call the onCreate() method after creating database and instantiate SQLiteDatabase object. Database name is passed in constructor call.

onUpgrade(SQLiteDatabase db,int oldVersion, int newVersion) is only called whenever there is a updation in existing version. So to update a version we have to increment the value of version variable passed in the superclass constructor.

In onUpgrade method we can write queries to perform whatever action is required. In most example you will see that existing table(s) are being dropped and again onCreate() method is being called to create tables again. But it's not mandatory to do so and it all depends upon your requirements.

We have to change database version if we have added a new row in the database table. If we have requirement that we don't want to lose existing data in the table then we can write alter table query in the onUpgrade(SQLiteDatabase db,int oldVersion, int newVersion) method.