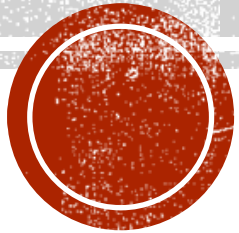
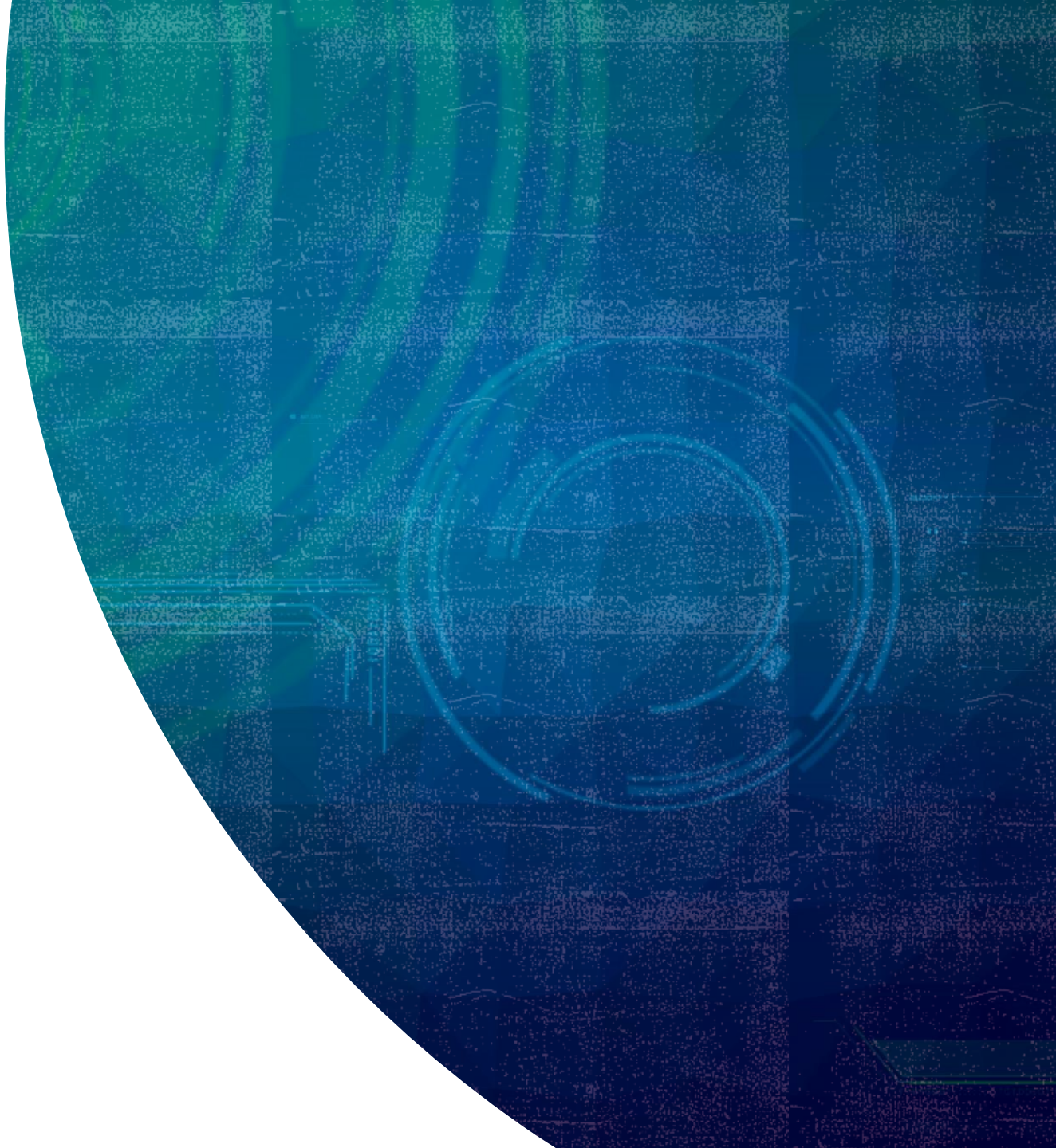
LECTURE #13
SQLITE IN ANDROID:



WHAT IS SQLITE:

SQLite is a popular software library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is highly reliable and used extensively in both large-scale and small-scale applications.

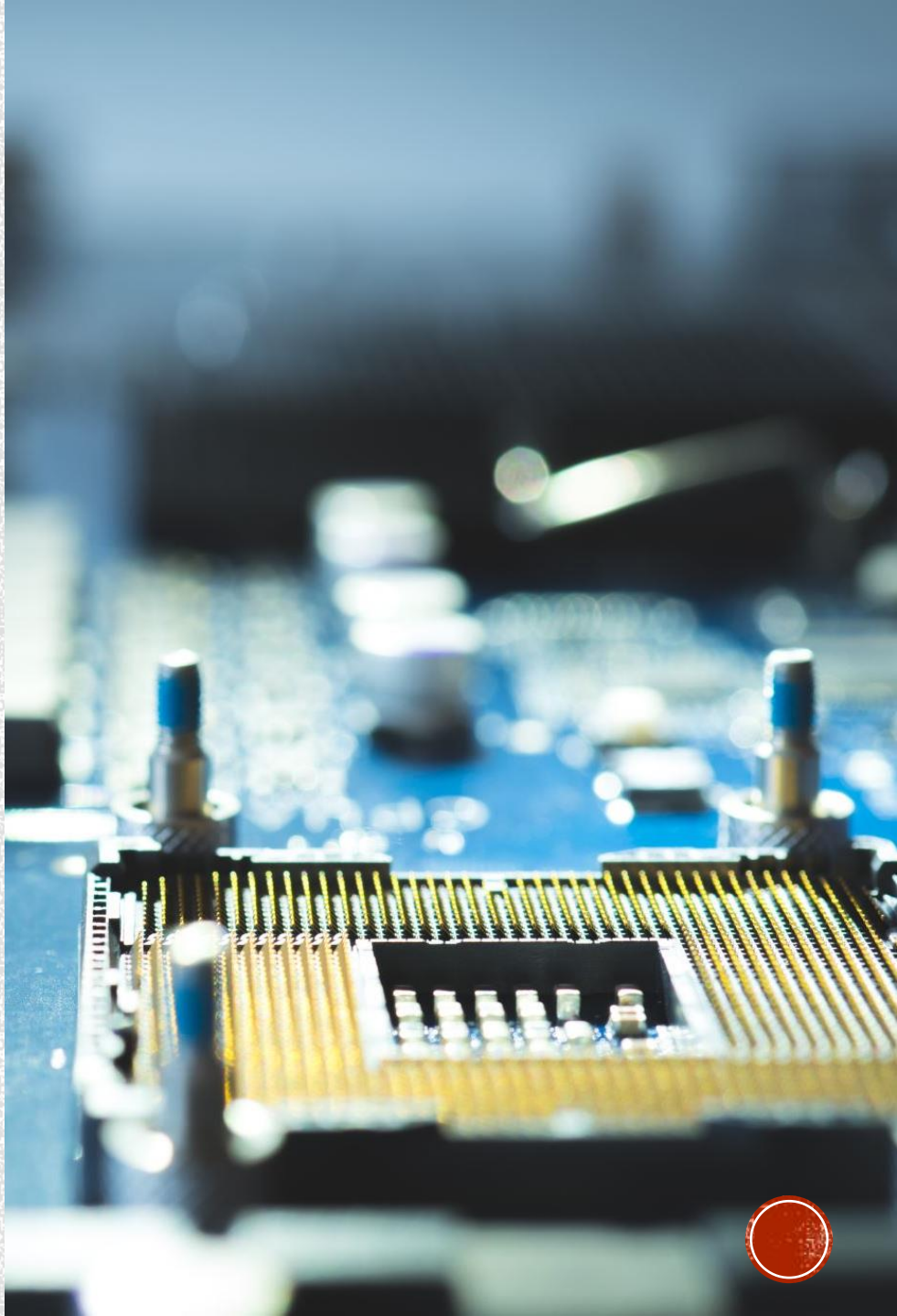


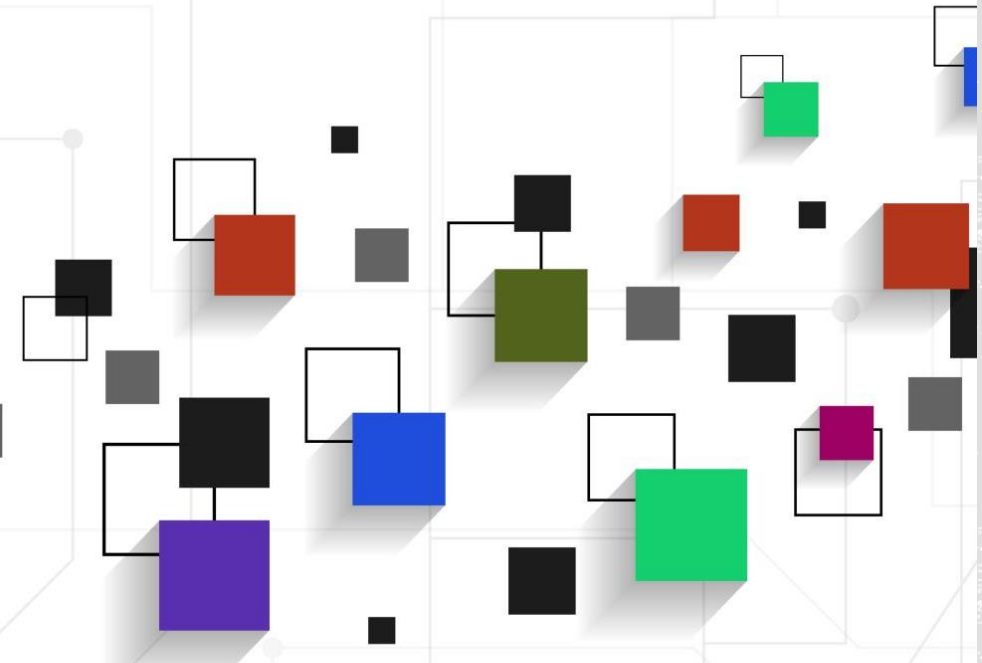
COMMON USES OF SQLITE:

- **Embedded Devices and Applications:** Due to its small size and easy integration, SQLite is ideal for use in embedded systems, mobile applications, personal electronics, and any device that requires internal data storage without the overhead of a server.

- **Web Browsers:** Many web browsers use SQLite for storing data such as history, cookies, and session information because of its lightweight and easy-to-integrate nature.

- **Data Analysis:** With support for SQL, it's also used for data analysis tasks. Data scientists and analysts might use SQLite to handle moderate-sized datasets due to its simplicity in setup and use.





HOW DATA STORED IN SQLITE:

Data is stored in the Android SQLite database in the form of tables. When we store this data in our SQLite database it is arranged in the form of tables that are similar to that of an Excel sheet.

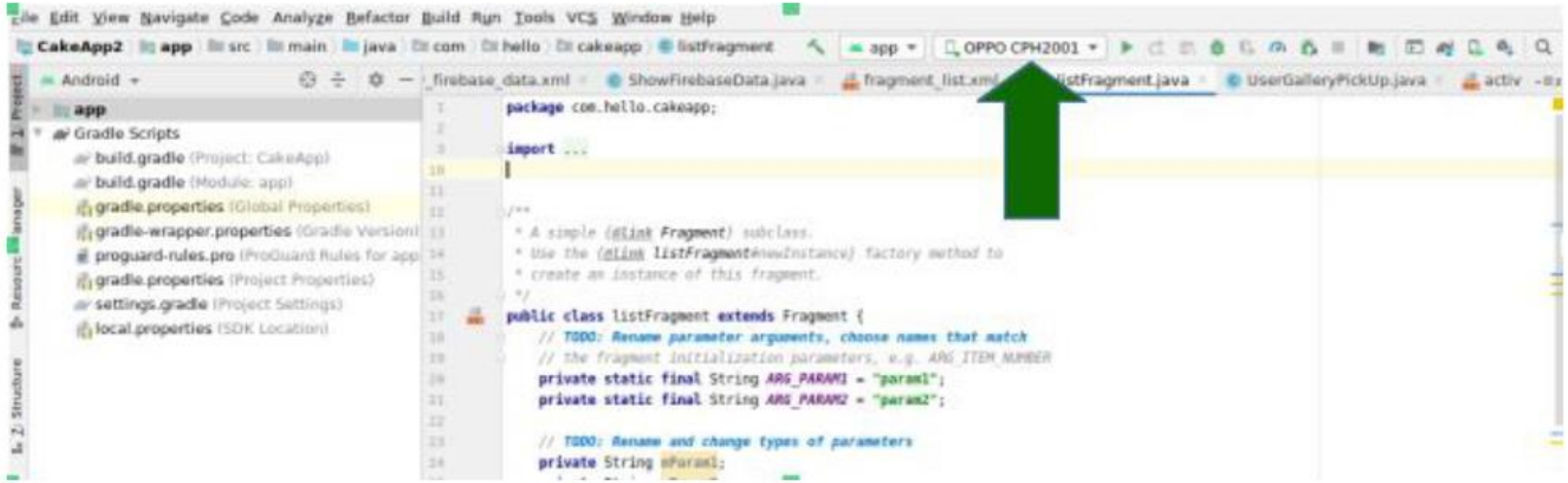
STEP TO VIEW DATABASE :

Step 1: Open android studio project which has SQLite database connection

CONNECT A DEVICE:

Step 2: Connect a device

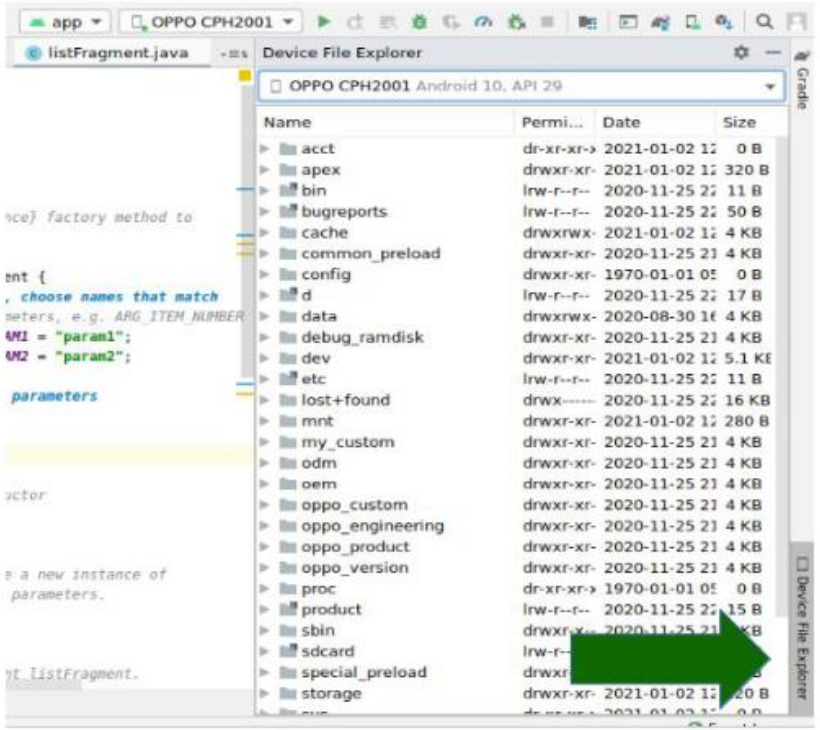
Connect external device or emulator with android studio make sure the name of the device must be shown on android studio screen.



STEP :3 SEARCH FOR FILE EXPLORER IN ANDROID STUDIO:

Step 3: Search for Device File Explorer in android studio

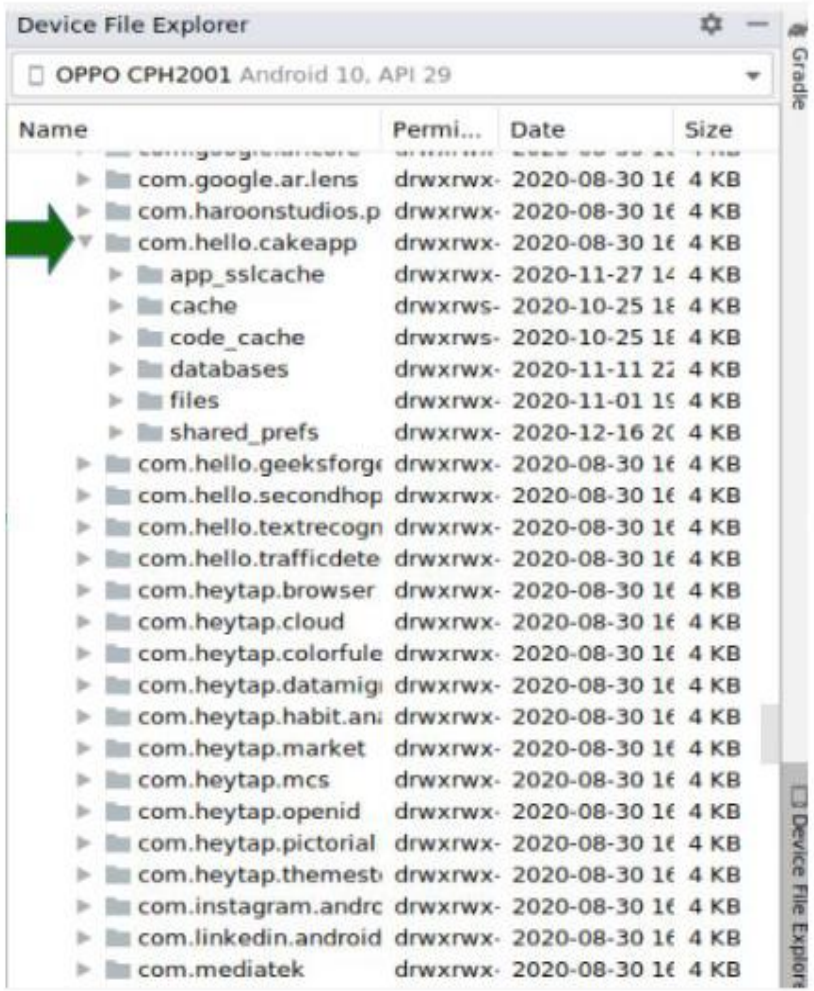
Device file explorer can be found in the bottom-right corner of the android studio screen. Click on **Device file explorer**.



STEP 4:

Step 4: Search application package name

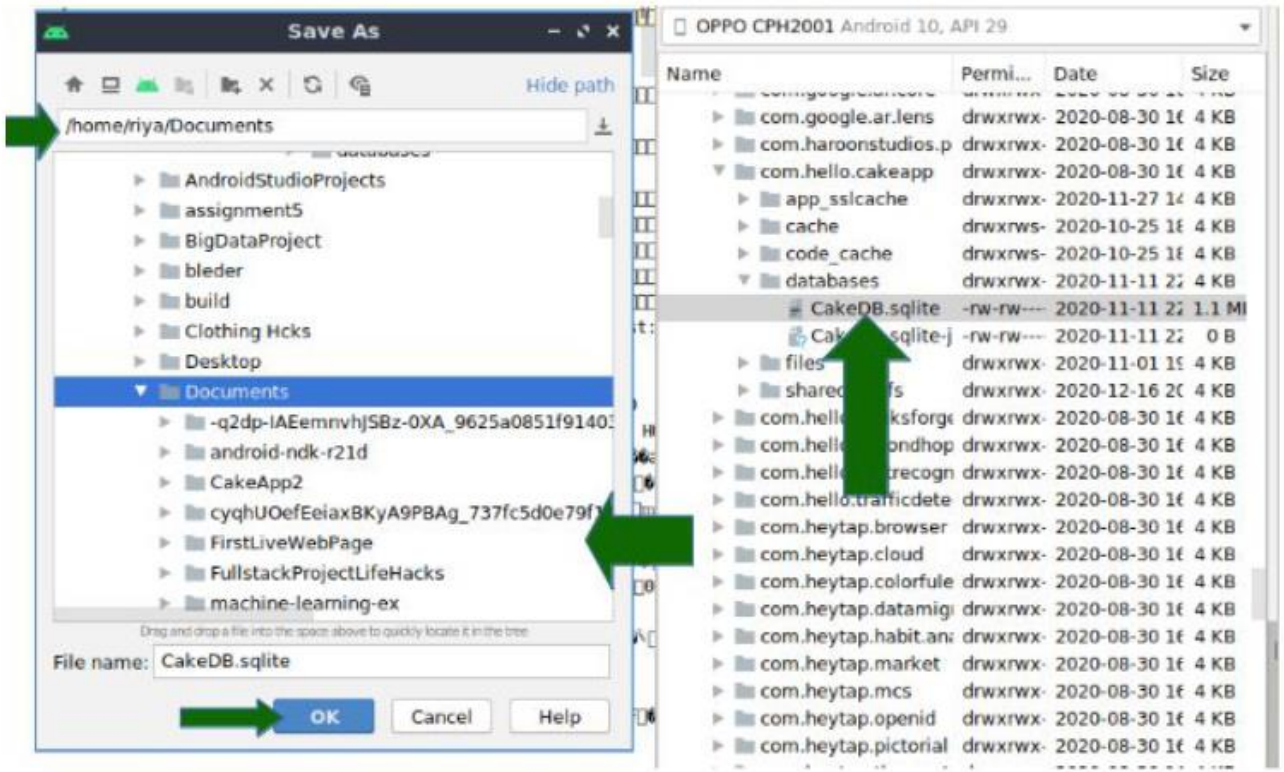
To search your package name go to **data > data> package name**. Click on package name.



STEP 5 :

Step 5: Download the database

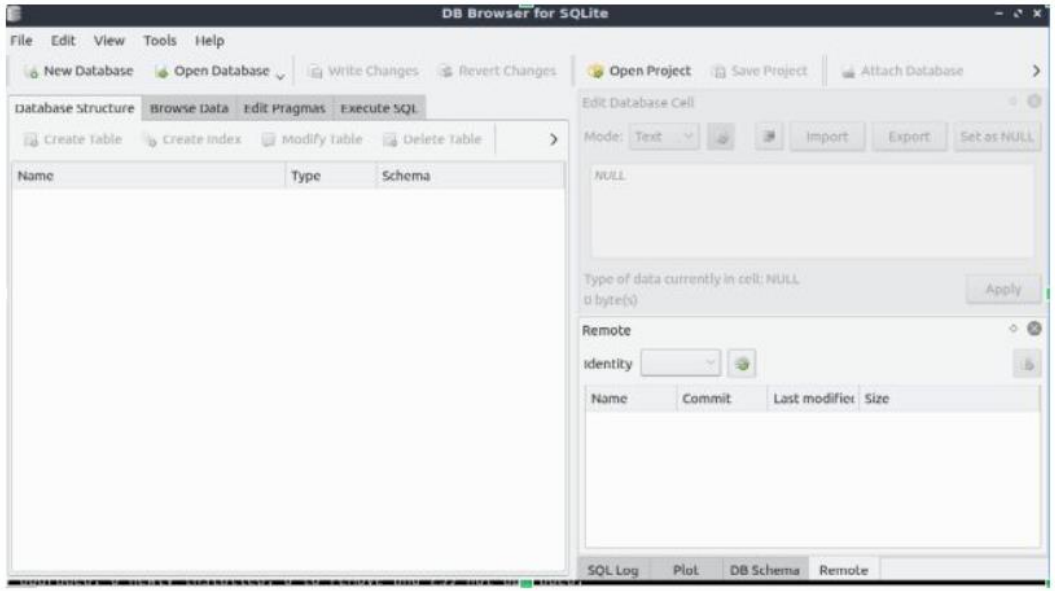
Now, select **database** and download database whose extension will be **.sqlite,** for that right-click on the database name and save file at any desired location but remember the location then click on **ok** in **Save As** dialog box.



STEP 6:

Step 6: Download SQLite browser

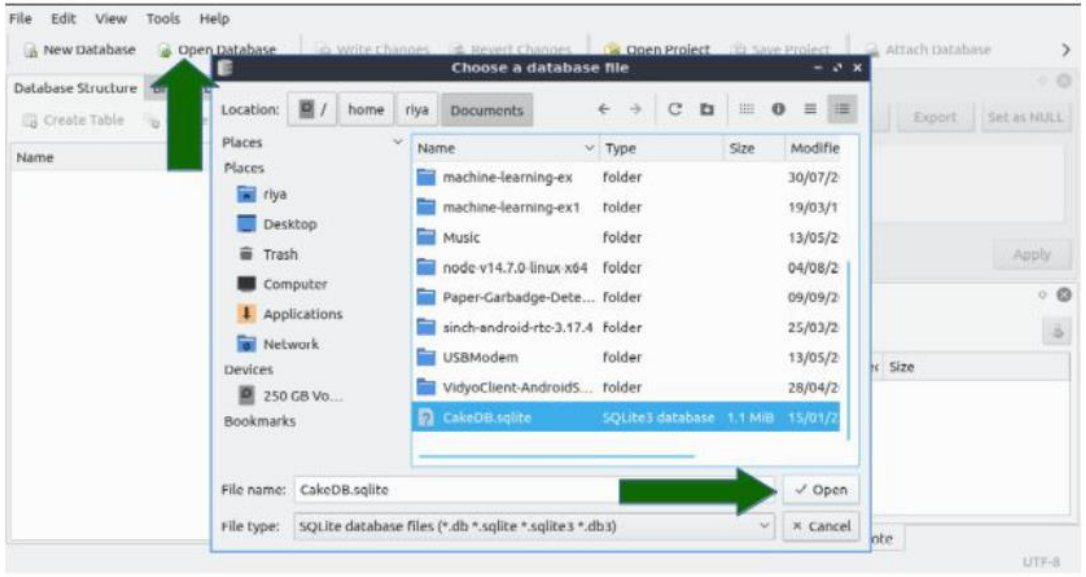
Now to view the database we required SQLite browser, you can download SQLite browser from https://sqlitebrowser.org/dl/. Download a suitable SQLite browser for your device from the above link and open it.



STEP : 7

Step 7: Search saved database file

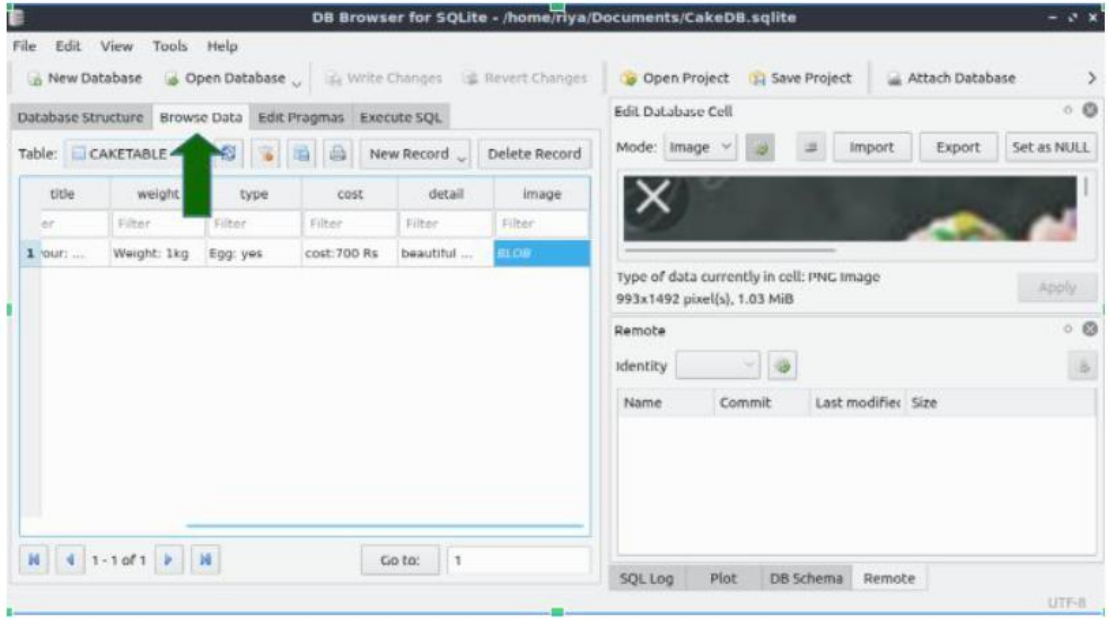
Click on the **open database** this will open a dialog box **choose a database file**. Now go to that location where you have saved the database previously and then select the database file and click on **open**.



STEP : 08

Step 8: View saved data in tables

To view data saved in the table click on **Browse data**, now that's it we have completed our today's task.



ACTIVITY.XML

hamza code

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <!--Edit text to enter course name-->
    <EditText
        android:id="@+id/idEdtCourseName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter course Name" />

    <!--edit text to enter course duration-->
    <EditText
        android:id="@+id/idEdtCourseDuration"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Duration" />

    <!--edit text to display course tracks-->
    <EditText
        android:id="@+id/idEdtCourseTracks"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Tracks" />

    <!--edit text for course description-->
    <EditText
        android:id="@+id/idEdtCourseDescription"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:hint="Enter Course Description" />

    <!--button for adding new course-->
    <Button
        android:id="@+id/idBtnAddCourse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:text="Add Course"
        android:textAllCaps="false" />

</LinearLayout>
```

CONTINUE:

```java
package org.hamza.sqliteinandroid;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    3 usages
    private EditText courseNameEdt, courseTracksEdt, courseDurationEdt, courseDescriptionEdt;
    2 usages
    private Button addCourseBtn;
    2 usages
    private DBHandler dbHandler;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        courseNameEdt = findViewById(R.id.idEdtCourseName);
        courseTracksEdt = findViewById(R.id.idEdtCourseTracks);
        courseDurationEdt = findViewById(R.id.idEdtCourseDuration);
        courseDescriptionEdt = findViewById(R.id.idEdtCourseDescription);
        addCourseBtn = findViewById(R.id.idBtnAddCourse);
```

CONTINUE:

activity_main.xml | MainActivity.java | DBHandler.java | AndroidManifest.xml

```
20          addCourseBtn = findViewById(R.id.idBtnAddCourse);
21          dbHandler = new DBHandler( context: MainActivity.this);
22          addCourseBtn.setOnClickListener(new View.OnClickListener() {
23              @Override
24              public void onClick(View v) {
25                  String courseName = courseNameEdt.getText().toString();
26                  String courseTracks = courseTracksEdt.getText().toString();
27                  String courseDuration = courseDurationEdt.getText().toString();
28                  String courseDescription = courseDescriptionEdt.getText().toString();
29                  if (courseName.isEmpty() && courseTracks.isEmpty() && courseDuration.isEmpty() && courseDescription.isEmpty()) {
30                      Toast.makeText( context: MainActivity.this, text: "Please enter all the data..", Toast.LENGTH_SHORT).show();
31                      return;
32                  }
33                  dbHandler.addNewCourse(courseName, courseDuration, courseDescription, courseTracks);
34                  Toast.makeText( context: MainActivity.this, text: "Course has been added.", Toast.LENGTH_SHORT).show();
35                  courseNameEdt.setText("");
36                  courseDurationEdt.setText("");
37                  courseTracksEdt.setText("");
38                  courseDescriptionEdt.setText("");
39              }
40          });
41      }
42  }
```

CONTINUE:

```
44                // validating if the text fields are empty or not.
45                if (courseName.isEmpty() && courseTracks.isEmpty() && courseDuration.isEmpty() && courseDescription.isEmpty()) {
46                    Toast.makeText( context: MainActivity.this,  text: "Please enter all the data..", Toast.LENGTH_SHORT).show();
47                    return;
48                }
49
50                // on below line we are calling a method to add new
51                // course to sqlite data and pass all our values to it.
52                dbHandler.addNewCourse(courseName, courseDuration, courseDescription, courseTracks);
53
54                // after adding the data we are displaying a toast message.
55                Toast.makeText( context: MainActivity.this,  text: "Course has been added.", Toast.LENGTH_SHORT).show();
56                courseNameEdt.setText("");
57                courseDurationEdt.setText("");
58                courseTracksEdt.setText("");
59                courseDescriptionEdt.setText("");
60            }
61        });
62    }
63}
```

IMPORT PACKAGES:

```
package org.hamza.sqliteinandroid;

import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
```

DB HANDLER CLASS

```
public class DBHandler extends SQLiteOpenHelper {

    1 usage
    private static final String DB_NAME = "coursedb";
    1 usage
    private static final int DB_VERSION = 1;
    3 usages
    private static final String TABLE_NAME = "mycourses";
    1 usage
    private static final String ID_COL = "id";
    2 usages
    private static final String NAME_COL = "name";
    2 usages
    private static final String DURATION_COL = "duration";
    2 usages
    private static final String DESCRIPTION_COL = "description";
    2 usages
    private static final String TRACKS_COL = "tracks";

    1 usage
    public DBHandler(Context context) { super(context, DB_NAME, factory: null, DB_VERSION); }
```

DB HANDLER .JAVA

activity_main.xml | MainActivity.java | DBHandler.java | AndroidManifest.xml

```
23      @Override
24      public void onCreate(SQLiteDatabase db) {
25          String query = "CREATE TABLE " + TABLE_NAME + " (" +
26                  ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
27                  NAME_COL + " TEXT," +
28                  DURATION_COL + " TEXT," +
29                  DESCRIPTION_COL + " TEXT," +
30                  TRACKS_COL + " TEXT)";
31          db.execSQL(query);
32      }
33
        1 usage
34      public void addNewCourse(String courseName, String courseDuration, String courseDescription, String courseTracks) {
35          SQLiteDatabase db = this.getWritableDatabase();
36          ContentValues values = new ContentValues();
37          values.put(NAME_COL, courseName);
38          values.put(DURATION_COL, courseDuration);
39          values.put(DESCRIPTION_COL, courseDescription);
40          values.put(TRACKS_COL, courseTracks);
41          db.insert(TABLE_NAME,  nullColumnHack: null, values);
42          db.close();
43      }
```

CONTINUE:

```
10 usages
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
    onCreate(db);
}
}
```

MANIFEST.XML

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.SQliteInAndroid"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
```

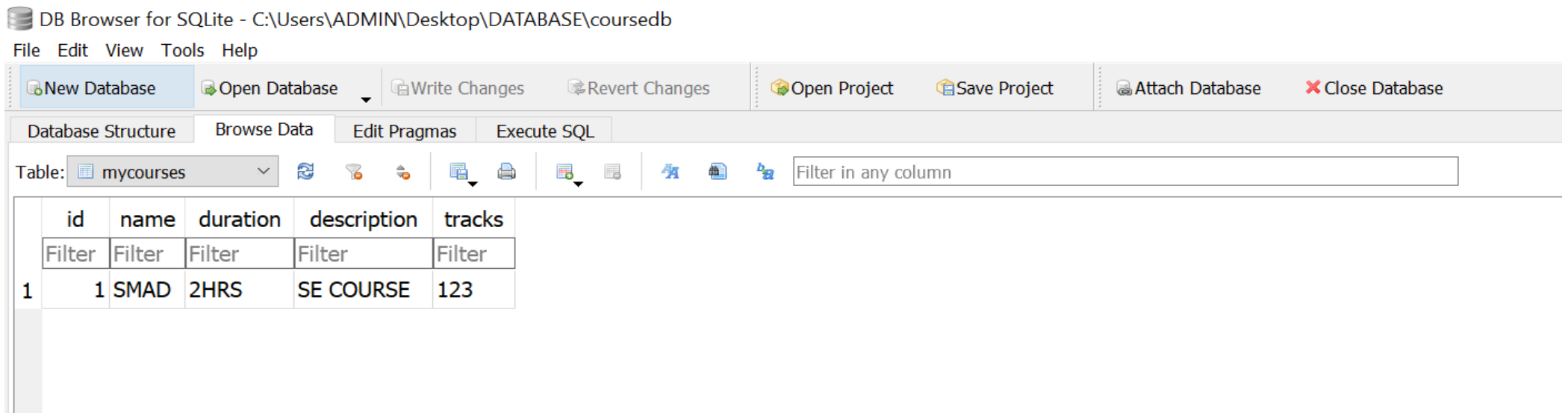
MANIFEST.XML

```
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />

</manifest>
```

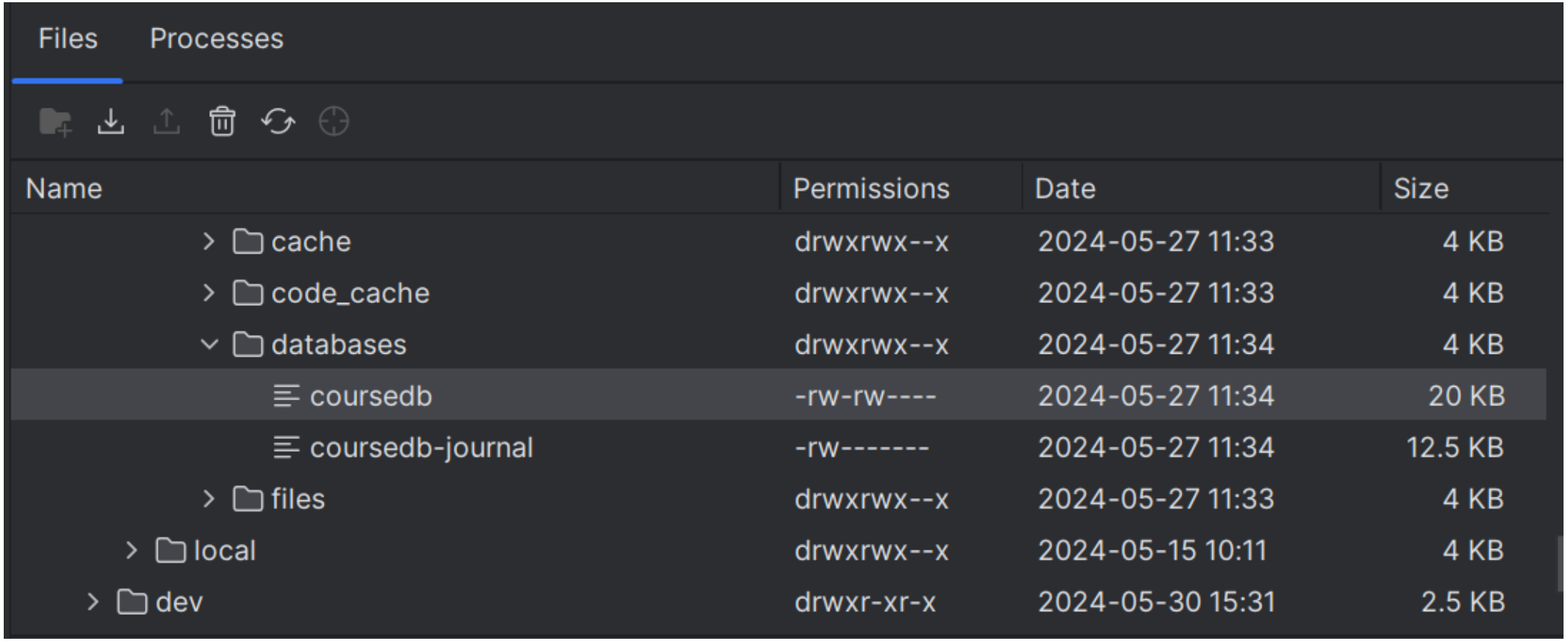
DATABASE VIEW WITH TABLE AND DATA:

Name	Type	Schema
Tables (3)		
android_metadata		CREATE TABLE android_metadata (locale TEXT)
mycourses		CREATE TABLE mycourses (id INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT,duration TEXT,description TEXT,tracks TEXT)
id	INTEGER	"id" INTEGER
name	TEXT	"name" TEXT
duration	TEXT	"duration" TEXT
description	TEXT	"description" TEXT
tracks	TEXT	"tracks" TEXT
sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)
Indices (0)		
Views (0)		
Triggers (0)		

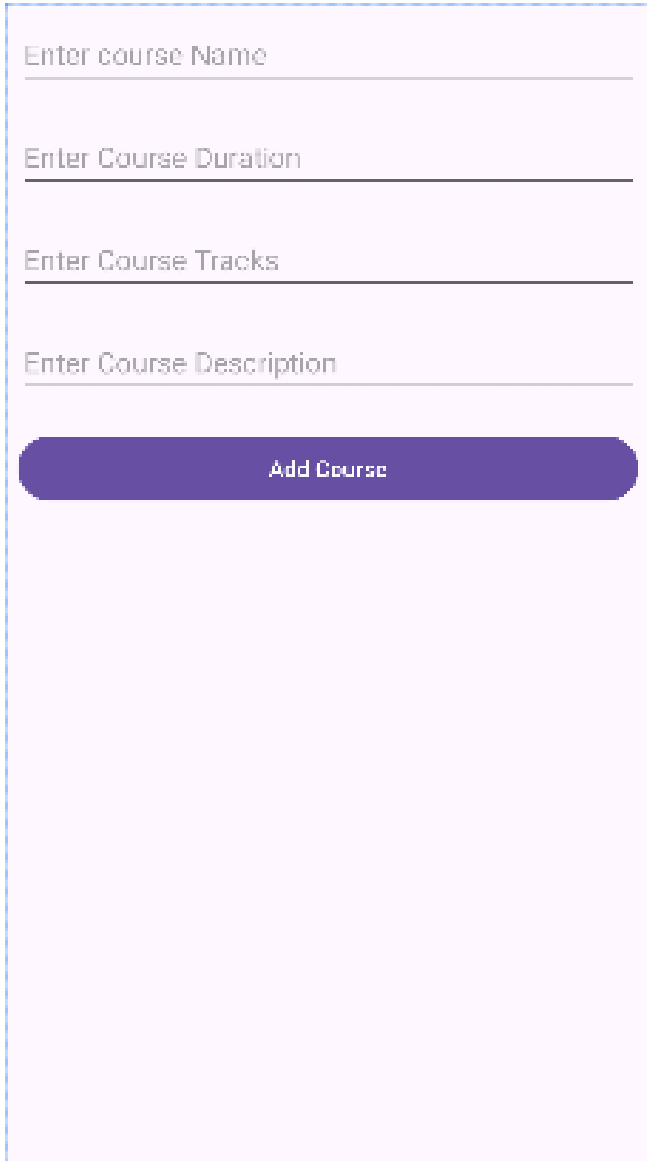


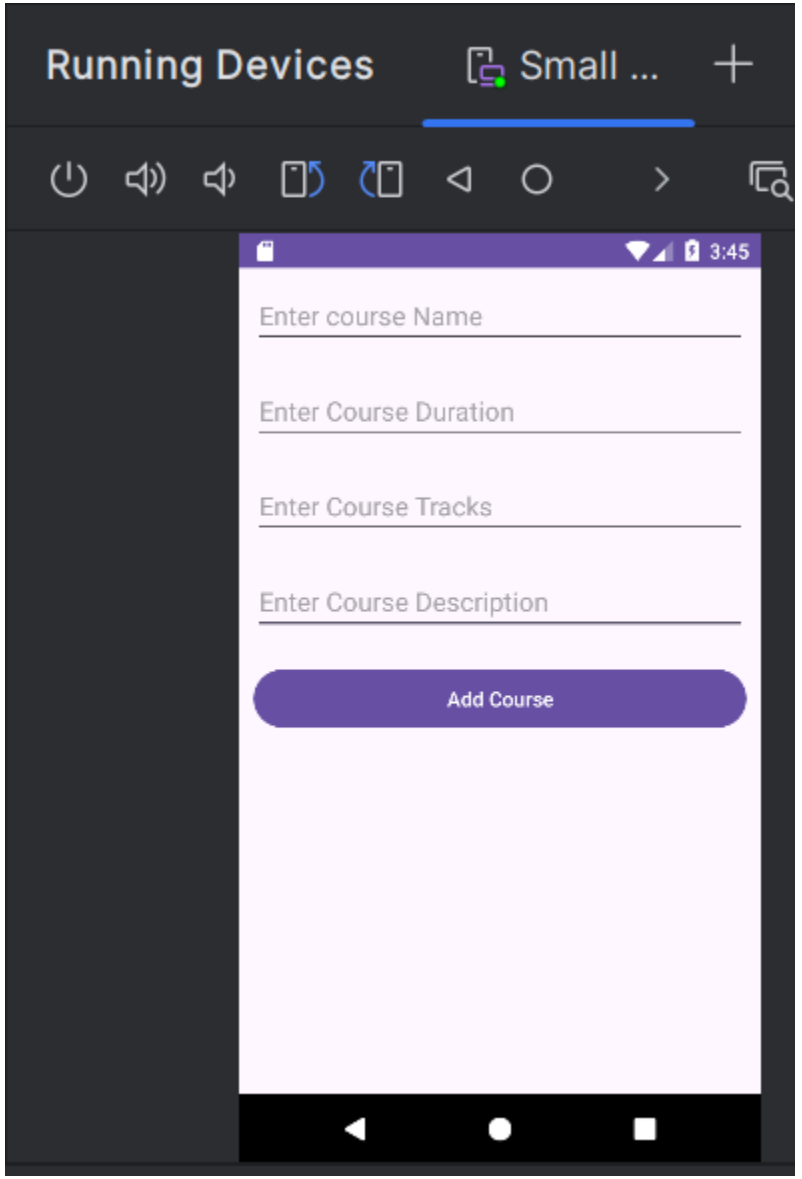
	id	name	duration	description	tracks
	Filter	Filter	Filter	Filter	Filter
1	1	SMAD	2HRS	SE COURSE	123

FILE LOCATION



OUTPUT:


Enter course Name
Enter Course Duration
Enter Course Tracks
Enter Course Description
Add Course


Running Devices
Small ...
3:45
Enter course Name
Enter Course Duration
Enter Course Tracks
Enter Course Description
Add Course

TASK : 1

In a basic E-commerce Android app, the app manages product and user data through a local SQLite database. It includes tables for **Users, Products, and Orders.** The Products table, for example, holds information like ID, name, and price. Users can browse products offline, add items to their cart, and place orders. The app syncs with a server when online to ensure data consistency. SQLite's lightweight design is ideal for fast data access and functionality without constant internet access, improving user experience in areas with unstable network conditions.