AT Lab – Database Connectivity using Android Studio

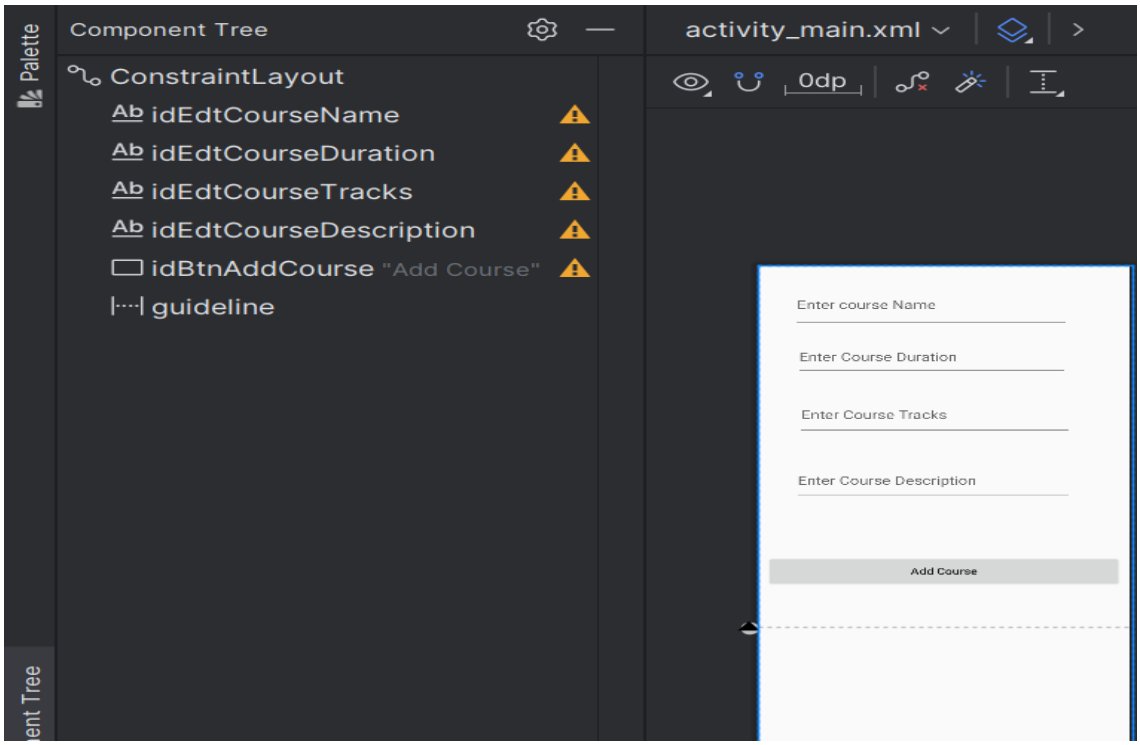
Step 1:

Navigate to the app > res > layout > activity_main.xml **and add the below code to that file. Below is the code for the** activity_main.xml **file.**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!--Edit text to enter course name-->

    <!--edit text to enter course duration-->

    <!--edit text to display course tracks-->

    <EditText
        android:id="@+id/idEdtCourseName"
        android:layout_width="304dp"
        android:layout_height="69dp"
        android:layout_margin="10dp"
        android:hint="Enter course Name"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.34"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.023" />

    <!--edit text for course description-->

    <EditText
        android:id="@+id/idEdtCourseDuration"
        android:layout_width="301dp"
        android:layout_height="57dp"
        android:layout_margin="10dp"
        android:layout_marginStart="16dp"
        android:hint="Enter Course Duration"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.35"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/idEdtCourseName"
        app:layout_constraintVertical_bias="0.012" />

    <!--button for adding new course-->
```

```
    <EditText
        android:id="@+id/idEdtCourseTracks"
        android:layout_width="303dp"
        android:layout_height="64dp"
        android:layout_margin="10dp"
        android:hint="Enter Course Tracks"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.388"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/idEdtCourseDuration"
        app:layout_constraintVertical_bias="0.038" />

    <EditText
        android:id="@+id/idEdtCourseDescription"
        android:layout_width="307dp"
        android:layout_height="63dp"
        android:layout_margin="10dp"
        android:layout_marginStart="8dp"
        android:hint="Enter Course Description"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.354"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/idEdtCourseTracks"
        app:layout_constraintVertical_bias="0.071" />

    <Button
        android:id="@+id/idBtnAddCourse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_marginStart="16dp"
        android:text="Add Course"
        android:textAllCaps="false"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/idEdtCourseDescription"
        app:layout_constraintVertical_bias="0.242" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="555dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Design of activity_main.xml:



Step 2: **Creating a new Java class for performing SQLite operations**

Navigate to the **app > java > your app's package name > Right-click on it > New > Java class** and name it as **DBHandler**

```
package com.example.myapplication_db1234;

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

public class DBHandler extends SQLiteOpenHelper {

    // creating a constant variables for our database.
    // below variable is for our database name.
    private static final String DB_NAME = "coursedb";

    // below int is our database version
    private static final int DB_VERSION = 1;

    // below variable is for our table name.
    private static final String TABLE_NAME = "mycourses";

    // below variable is for our id column.
    private static final String ID_COL = "id";

    // below variable is for our course name column
    private static final String NAME_COL = "name";
```

```
    // below variable id for our course duration column.
    private static final String DURATION_COL = "duration";

    // below variable for our course description column.
    private static final String DESCRIPTION_COL = "description";

    // below variable is for our course tracks column.
    private static final String TRACKS_COL = "tracks";

    // creating a constructor for our database handler.
    public DBHandler(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
    }

    // below method is for creating a database by running a sqlite query
    @Override
    public void onCreate(SQLiteDatabase db) {
        // on below line we are creating
        // an sqlite query and we are
        // setting our column names
        // along with their data types.
        String query = "CREATE TABLE " + TABLE_NAME + " ("
                + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + NAME_COL + " TEXT,"
                + DURATION_COL + " TEXT,"
                + DESCRIPTION_COL + " TEXT,"
                + TRACKS_COL + " TEXT)";

        // at last we are calling a exec sql
        // method to execute above sql query
        db.execSQL(query);
    }

    // this method is use to add new course to our sqlite database.
    public void addNewCourse(String courseName, String courseDuration, String
courseDescription, String courseTracks) {

        // on below line we are creating a variable for
        // our sqlite database and calling writable method
        // as we are writing data in our database.
        SQLiteDatabase db = this.getWritableDatabase();

        // on below line we are creating a
        // variable for content values.
        ContentValues values = new ContentValues();

        // on below line we are passing all values
        // along with its key and value pair.
        values.put(NAME_COL, courseName);
        values.put(DURATION_COL, courseDuration);
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);
```

```
        // after adding all values we are passing
        // content values to our table.
        db.insert(TABLE_NAME, null, values);

        // at last we are closing our
        // database after adding database.
        db.close();
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // this method is called to check if the table exists already.
        db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
        onCreate(db);
    }
}
```

Step 3: **Working with the MainActivity.java file**

Go to the **MainActivity.java** file:

```
package com.example.myapplication_db1234;

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 {

    // creating variables for our edittext, button and dbhandler
    private EditText courseNameEdt, courseTracksEdt, courseDurationEdt,
courseDescriptionEdt;
    private Button addCourseBtn;
    private DBHandler dbHandler;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // initializing all our variables.
        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);
```

```
        // creating a new dbhandler class
        // and passing our context to it.
        dbHandler = new DBHandler(MainActivity.this);

        // below line is to add on click listener for our add course button.
        addCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

                // below line is to get data from all edit text fields.
                String courseName = courseNameEdt.getText().toString();
                String courseTracks = courseTracksEdt.getText().toString();
                String courseDuration = courseDurationEdt.getText().toString();
                String courseDescription = courseDescriptionEdt.getText().toString();

                // validating if the text fields are empty or not.
                if (courseName.isEmpty() && courseTracks.isEmpty() &&
courseDuration.isEmpty() && courseDescription.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please enter all the data..",
Toast.LENGTH_SHORT).show();
                    return;
                }

                // on below line we are calling a method to add new
                // course to sqlite data and pass all our values to it.
                dbHandler.addNewCourse(courseName, courseDuration, courseDescription,
courseTracks);

                // after adding the data we are displaying a toast message.
                Toast.makeText(MainActivity.this, "Course has been added.",
Toast.LENGTH_SHORT).show();
                courseNameEdt.setText("");
                courseDurationEdt.setText("");
                courseTracksEdt.setText("");
                courseDescriptionEdt.setText("");
            }
        });
    }
}
```

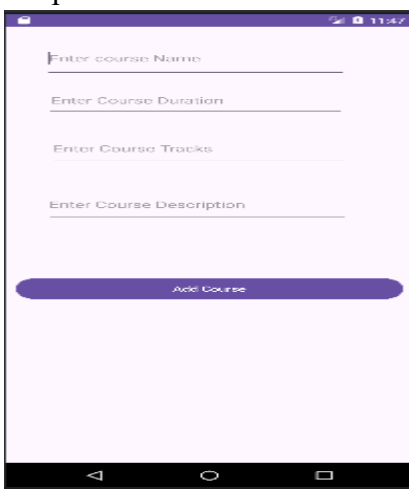
Step 4: AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

```
<!--Edit text to enter course name-->

<!--edit text to enter course duration-->

<!--edit text to display course tracks-->

<EditText
    android:id="@+id/idEdtCourseName"
    android:layout_width="304dp"
    android:layout_height="69dp"
    android:layout_margin="10dp"
    android:hint="Enter course Name"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.34"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.023" />

<!--edit text for course description-->

<EditText
    android:id="@+id/idEdtCourseDuration"
    android:layout_width="301dp"
    android:layout_height="57dp"
    android:layout_margin="10dp"
    android:layout_marginStart="16dp"
    android:hint="Enter Course Duration"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.35"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/idEdtCourseName"
    app:layout_constraintVertical_bias="0.012" />

<!--button for adding new course-->

<EditText
    android:id="@+id/idEdtCourseTracks"
    android:layout_width="303dp"
    android:layout_height="64dp"
    android:layout_margin="10dp"
    android:hint="Enter Course Tracks"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.388"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/idEdtCourseDuration"
    app:layout_constraintVertical_bias="0.038" />

<EditText
    android:id="@+id/idEdtCourseDescription"
```

```
        android:layout_width="307dp"
        android:layout_height="63dp"
        android:layout_margin="10dp"
        android:layout_marginStart="8dp"
        android:hint="Enter Course Description"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.354"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/idEdtCourseTracks"
        app:layout_constraintVertical_bias="0.071" />

    <Button
        android:id="@+id/idBtnAddCourse"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_marginStart="16dp"
        android:text="Add Course"
        android:textAllCaps="false"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.0"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/idEdtCourseDescription"
        app:layout_constraintVertical_bias="0.242" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="555dp" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Output:

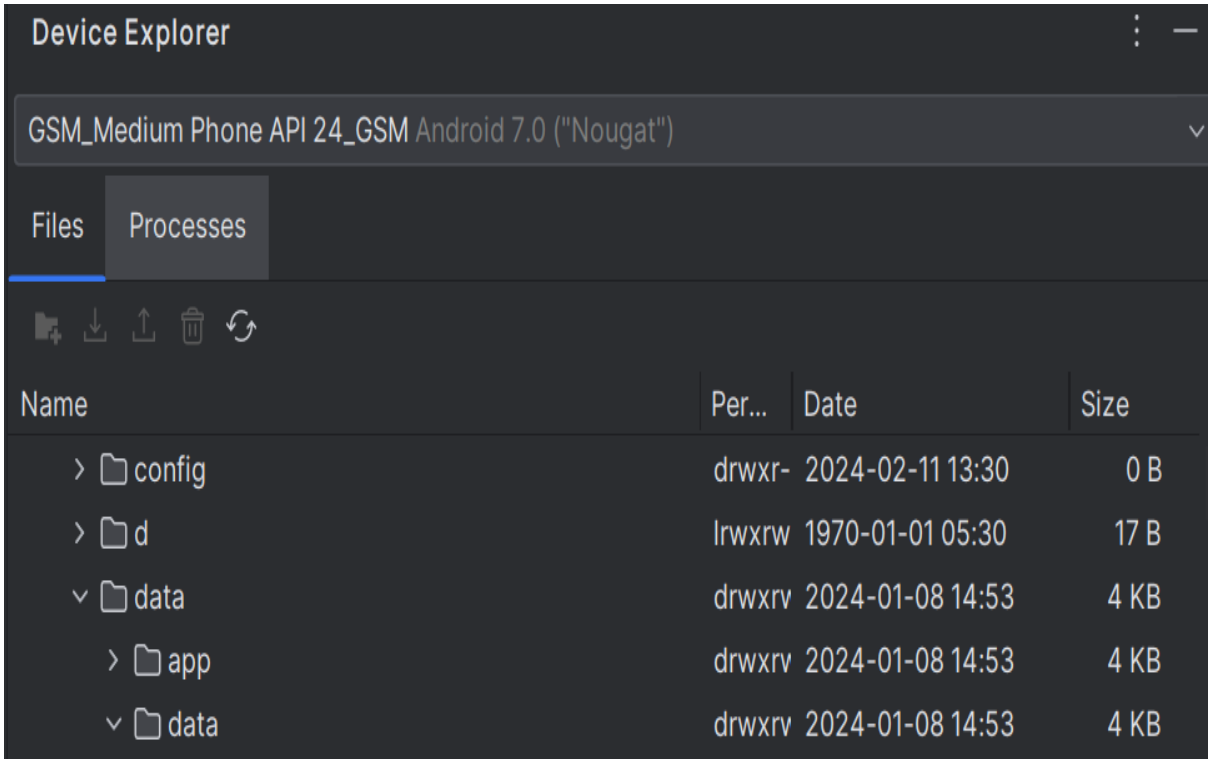


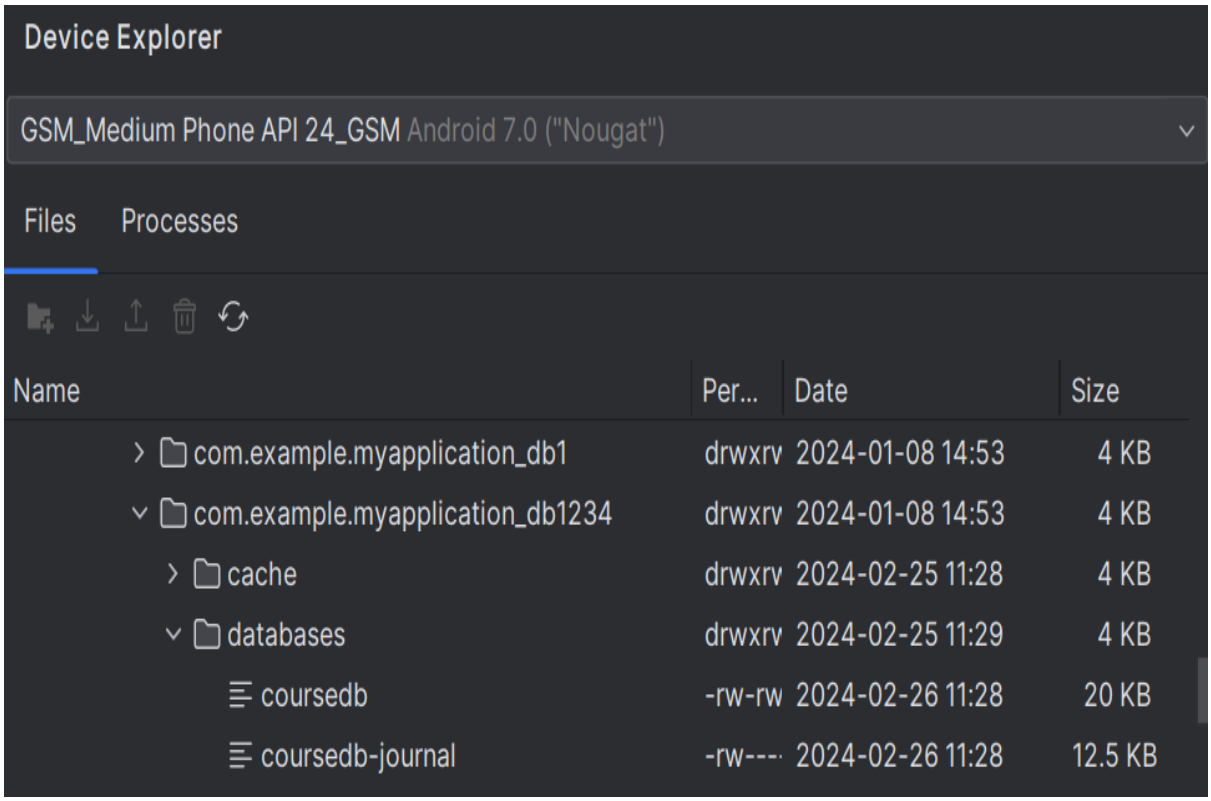
Input 4-5 data in the fields.

How to View and Locate SQLite Database in Android Studio?

Step 5: Search for Device File Explorer in android studio

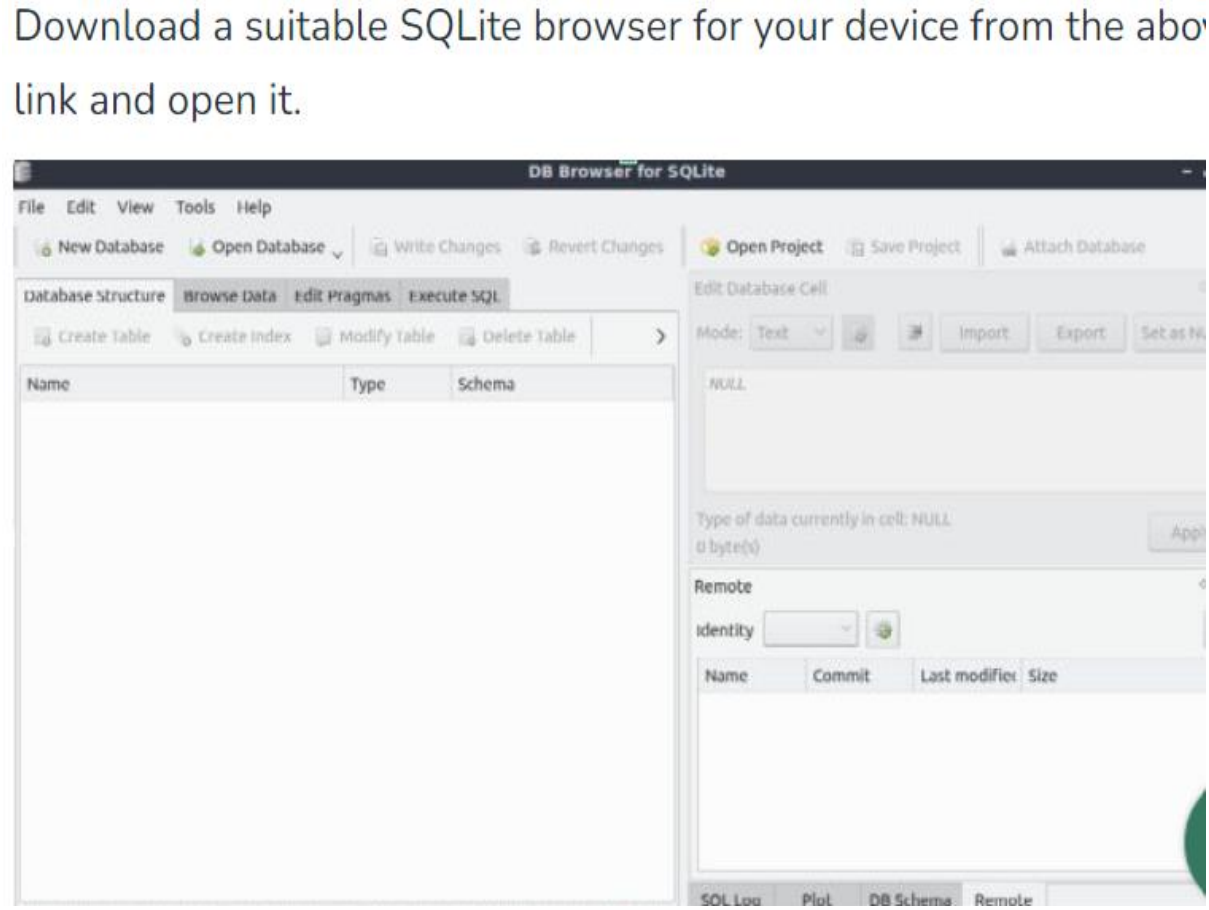
Device file explorer can be found in the bottom-right corner of the android studio screen. Click on Device file explorer.





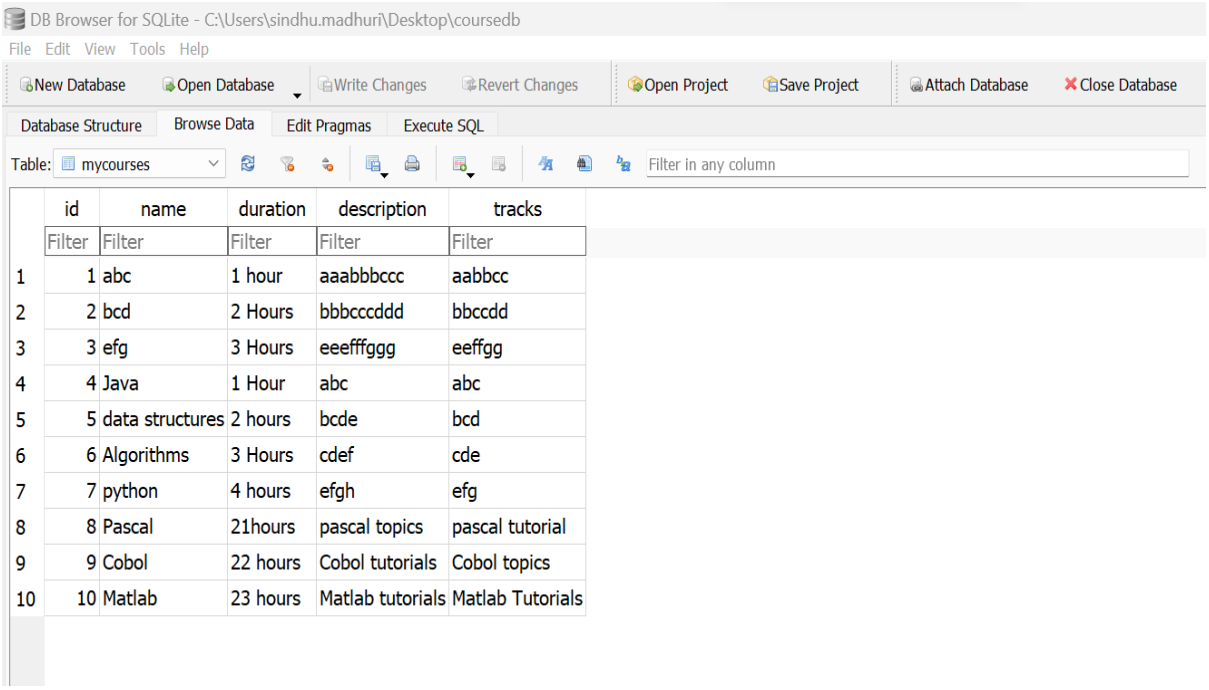
Step 6: Download SQLite browser

Now to view the database we required SQLite browser, you can download SQLite browser from https://sqlitebrowser.org/dl/.



Step 7: Search saved database file & view saved data in tables:

To view data saved in the table click on **Browse data.**



	id	name	duration	description	tracks
	Filter	Filter	Filter	Filter	Filter
1	1	abc	1 hour	aaabbbccc	aabbcc
2	2	bcd	2 Hours	bbbcccddd	bbccdd
3	3	efg	3 Hours	eeefffggg	eeffgg
4	4	Java	1 Hour	abc	abc
5	5	data structures	2 hours	bcde	bcd
6	6	Algorithms	3 Hours	cdef	cde
7	7	python	4 hours	efgh	efg
8	8	Pascal	21hours	pascal topics	pascal tutorial
9	9	Cobol	22 hours	Cobol tutorials	Cobol topics
10	10	Matlab	23 hours	Matlab tutorials	Matlab Tutorials