Lab 21: Android SQLite Database

Introduction

For tasks like storing, altering, or retrieving persistent data from the database on Android devices, SQLite is an open-source relational database. It comes pre-installed on Android. Therefore, no database setup or management tasks are required. The ability to use the SQLite database is provided by the SQLiteOpenHelper class.

Let's get Started:

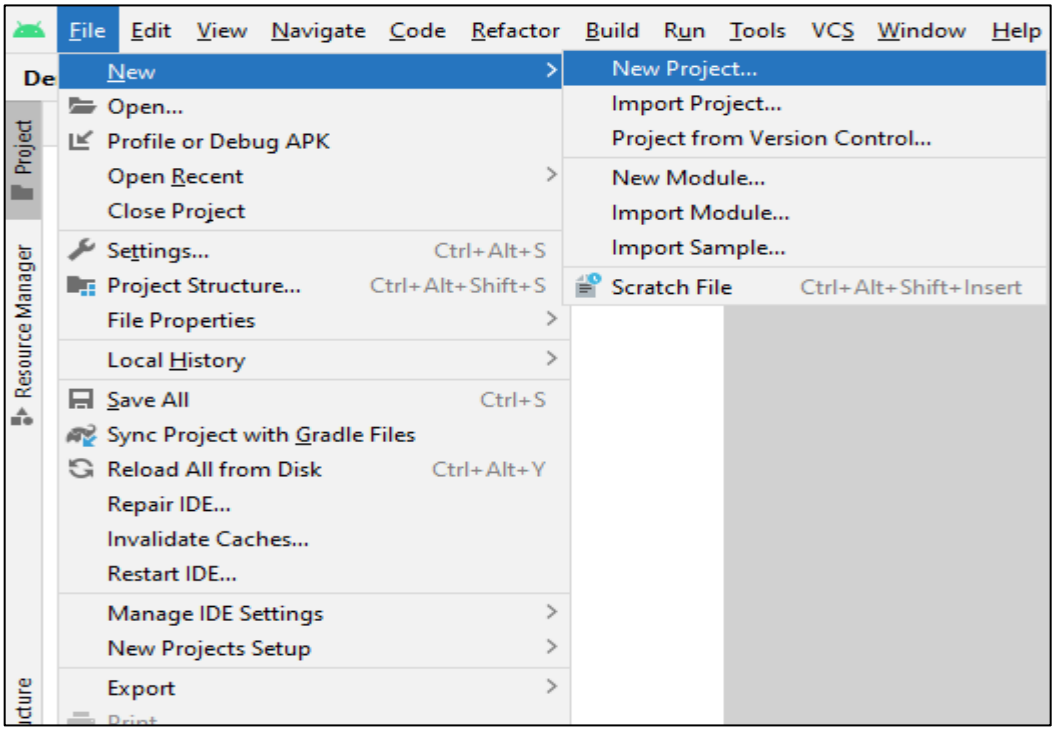
In this experiment we will develop an Android App to demonstrate the use of Android SQLite Database.

- Launching File Explorer
- access the data directory
- Look up the name of your application package in the data directory.
- Go to databases in your application package to access your database (contactsManager)
- A copy of your database can be saved.
- any tool or browser extension for SQLite. DB Browser for SQLite, for instance
- Open your database in the programme (DB Browser for SQLite) by launching it.
- You can then choose and view the data in your database depending on the tool you're using.
- To see the stored data, for instance, pick your table (contacts) from the Browse Data menu in the DB Browser for SQLite.

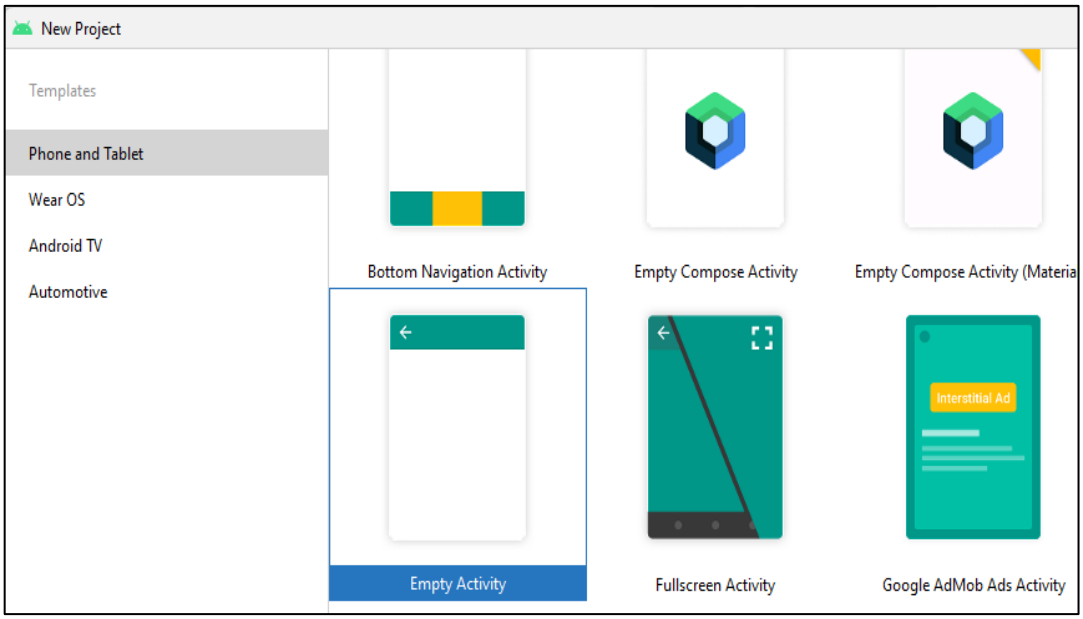
Download & Install

- DB Browser for SQLite - Standard installer for 64-bit Windows

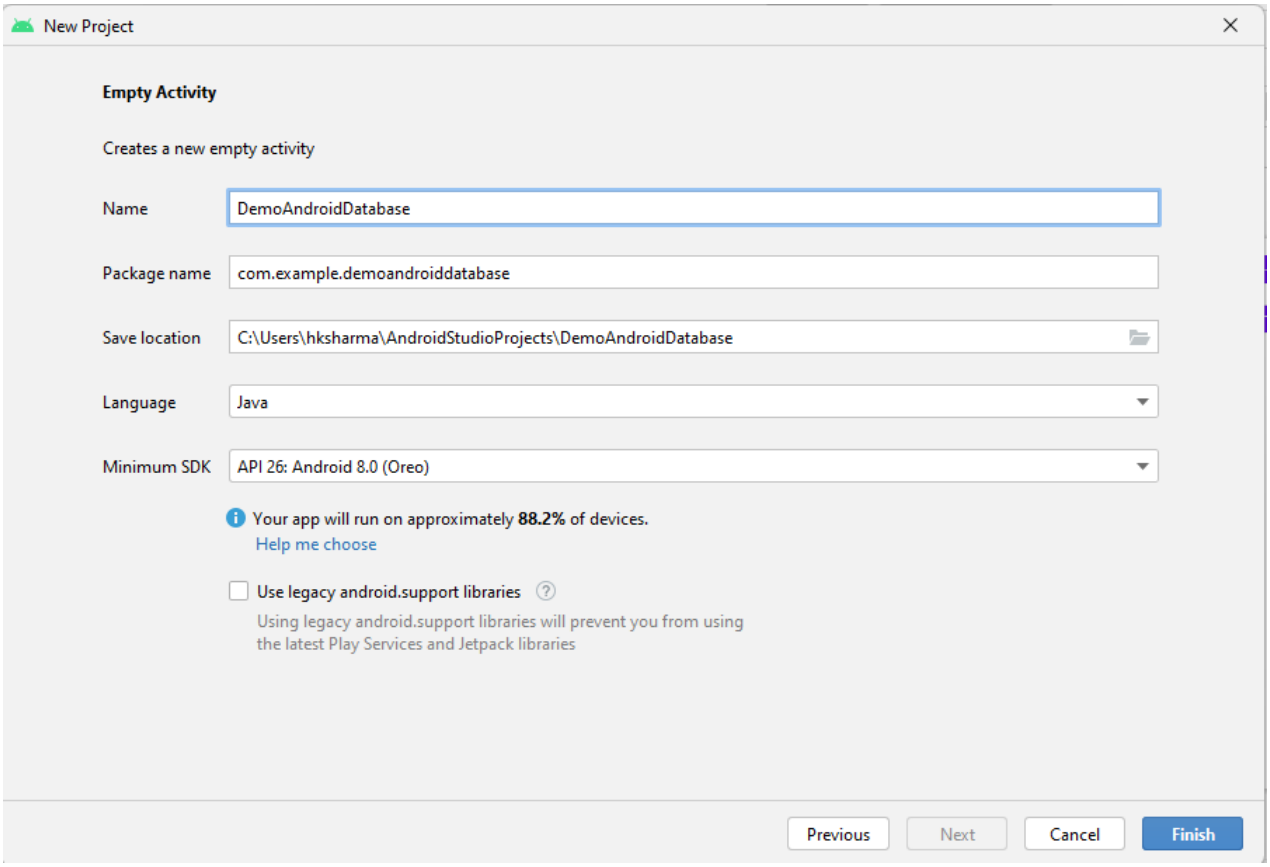
Step 1: Create a New Project in Android Studio as shown below



Step 2: Select Empty Activity as shown below



Step 3: Provide a Project Name as shown below



Step 4: Update MainActivity.java as per the code given below

```
package com.example.demosql23;
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);
        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);
        dbHandler = new DBHandler(MainActivity.this);
        addCourseBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String courseName = courseNameEdt.getText().toString();
                String courseTracks = courseTracksEdt.getText().toString();
                String courseDuration =
courseDurationEdt.getText().toString();
                String courseDescription =
courseDescriptionEdt.getText().toString();
                if (courseName.isEmpty() && courseTracks.isEmpty() &&
courseDuration.isEmpty() && courseDescription.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please enter all the
data..", Toast.LENGTH_SHORT).show();
                    return;
                }
                dbHandler.addNewCourse(courseName, courseDuration,
courseDescription, courseTracks);
                 Toast.makeText(MainActivity.this, "Course has been added.",
Toast.LENGTH_SHORT).show();
                courseNameEdt.setText("");
                courseDurationEdt.setText("");
                courseTracksEdt.setText("");
                courseDescriptionEdt.setText("");
            }
        });
    }
}
```

Step 5: Create DBHandler.java as per the code given below

```
package com.example.demosql23;
import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class DBHandler extends SQLiteOpenHelper {
    private static final String DB_NAME = "coursedb";
    private static final int DB_VERSION = 1;
    private static final String TABLE_NAME = "mycourses";
    private static final String ID_COL = "id";
    private static final String NAME_COL = "name";
    private static final String DURATION_COL = "duration";
    private static final String DESCRIPTION_COL = "description";
    private static final String TRACKS_COL = "tracks";
    public DBHandler(Context context) {
        super(context, DB_NAME, null, DB_VERSION);
```

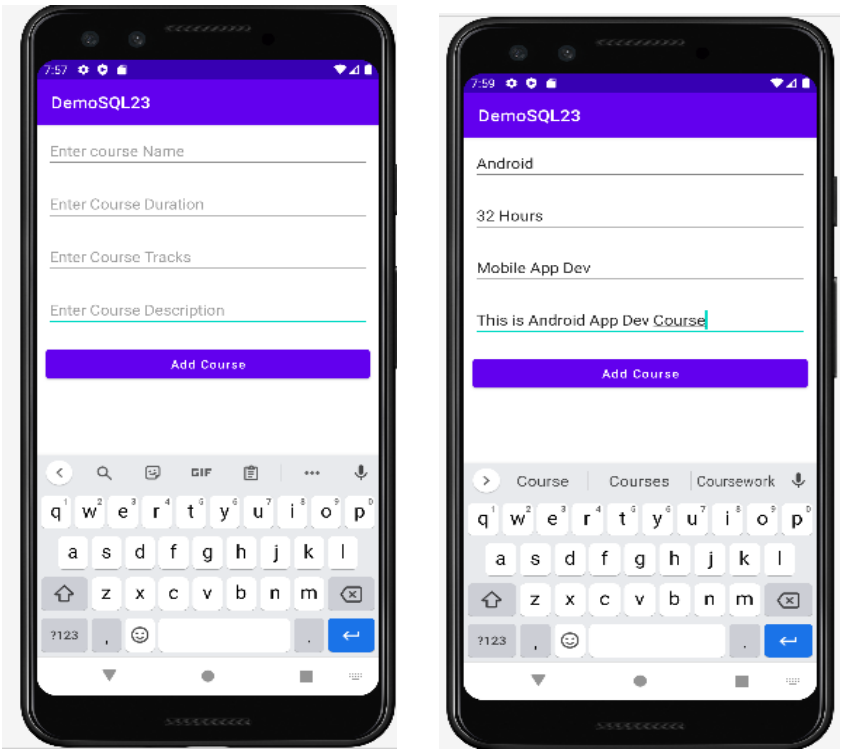
```
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        String query = "CREATE TABLE " + TABLE_NAME + " ("
                + ID_COL + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + NAME_COL + " TEXT,"
                + DURATION_COL + " TEXT,"
                + DESCRIPTION_COL + " TEXT,"
                + TRACKS_COL + " TEXT)";
        db.execSQL(query);
    }
    public void addNewCourse(String courseName, String courseDuration, String
courseDescription, String courseTracks) {

        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues values = new ContentValues();
        values.put(NAME_COL, courseName);
        values.put(DURATION_COL, courseDuration);
        values.put(DESCRIPTION_COL, courseDescription);
        values.put(TRACKS_COL, courseTracks);
        db.insert(TABLE_NAME, null, values);
        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 6: Update activity_main.xml for Relative Layout as per the code given below

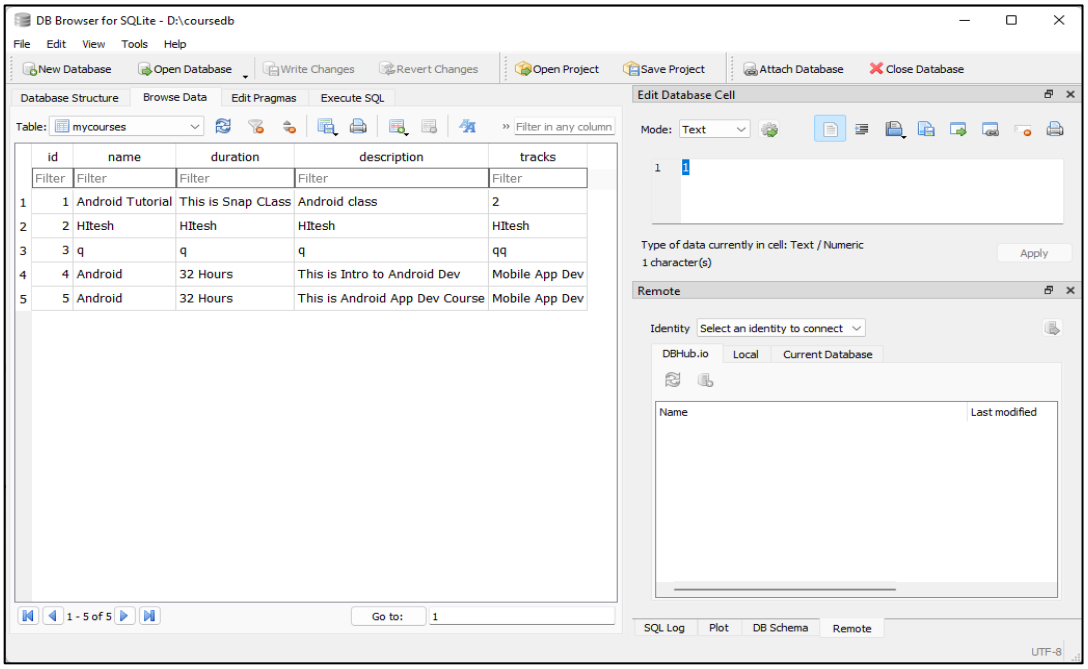
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <!--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>
```

Step 7: Check Output on Android Emulator and it should look like as given below



Step 8: Check Database as given below



Voila!! We have successfully completed this lab.