

Activity_main.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"
    android:padding="16dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/etRegisterNumber"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Register Number"
        android:inputType="number" />

    <EditText
        android:id="@+id/etName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Name" />

    <EditText
        android:id="@+id/etCGPA"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="CGPA"
        android:inputType="numberDecimal" />

    <Button
        android:id="@+id/btnAdd"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Add" />

    <Button
        android:id="@+id/btnModify"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Modify" />

    <Button
        android:id="@+id/btnDelete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Delete" />

    <Button
        android:id="@+id/btnView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="View" />

    <Button
        android:id="@+id/btnClear"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Clear" />

    <TextView
        android:text=""
        android:textSize="15dp"

```

```
        android:layout_width="376dp"
        android:layout_height="308dp" android:id="@+id/tvDetails"/>
```

```
</LinearLayout>
```

MainActivity.kt

```
package com.example.ex5

import UsersDBHelper
import android.content.ContentValues
import android.content.Context
import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.os.Bundle
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast
import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

    private lateinit var etRegisterNumber: EditText
    private lateinit var etName: EditText
    private lateinit var etCGPA: EditText
    private lateinit var btnAdd: Button
    private lateinit var btnModify: Button
    private lateinit var btnDelete: Button
    private lateinit var btnView: Button
    private lateinit var btnClear: Button
    private lateinit var tvDetails: TextView

    private lateinit var dbHelper: UsersDBHelper

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        dbHelper = UsersDBHelper(this)

        etRegisterNumber = findViewById(R.id.etRegisterNumber)
        etName = findViewById(R.id.etName)
        etCGPA = findViewById(R.id.etCGPA)
        btnAdd = findViewById(R.id.btnAdd)
        btnModify = findViewById(R.id.btnModify)
        btnDelete = findViewById(R.id.btnDelete)
        btnView = findViewById(R.id.btnView)
        btnClear = findViewById(R.id.btnClear)
        tvDetails = findViewById(R.id.tvDetails)

        btnAdd.setOnClickListener {
            insertData()
        }

        btnModify.setOnClickListener {
            updateData()
        }

        btnDelete.setOnClickListener {
            deleteData()
        }
    }
}
```

```

        btnView.setOnClickListener {
            viewData()
        }

        btnClear.setOnClickListener {
            clearFields()
        }
    }

    private fun insertData() {
        val registerNumber = etRegisterNumber.text.toString().toInt()
        val name = etName.text.toString()
        val cgpa = etCGPA.text.toString().toDouble()

        val db = dbHelper.writableDatabase

        val values = ContentValues().apply {
            put(DBContract.StudentEntry.COLUMN_REGISTER_NUMBER, registerNumber)
            put(DBContract.StudentEntry.COLUMN_NAME, name)
            put(DBContract.StudentEntry.COLUMN_CGPA, cgpa)
        }

        val newRowId = db?.insert(DBContract.StudentEntry.TABLE_NAME, null, values)

        Toast.makeText(this, "Inserted Row ID: $newRowId",
            Toast.LENGTH_SHORT).show()
    }

    private fun updateData() {
        val registerNumber = etRegisterNumber.text.toString().toInt()
        val name = etName.text.toString()
        val cgpa = etCGPA.text.toString().toDouble()

        val db = dbHelper.writableDatabase

        val values = ContentValues().apply {
            put(DBContract.StudentEntry.COLUMN_NAME, name)
            put(DBContract.StudentEntry.COLUMN_CGPA, cgpa)
        }

        val selection = "${DBContract.StudentEntry.COLUMN_REGISTER_NUMBER} = ?"
        val selectionArgs = arrayOf(registerNumber.toString())

        val count = db?.update(
            DBContract.StudentEntry.TABLE_NAME,
            values,
            selection,
            selectionArgs
        )

        Toast.makeText(this, "Updated $count rows", Toast.LENGTH_SHORT).show()
    }

    private fun deleteData() {
        val registerNumber = etRegisterNumber.text.toString().toInt()

        val db = dbHelper.writableDatabase

        val selection = "${DBContract.StudentEntry.COLUMN_REGISTER_NUMBER} = ?"
        val selectionArgs = arrayOf(registerNumber.toString())

        val deletedRows = db?.delete(DBContract.StudentEntry.TABLE_NAME, selection,
            selectionArgs)
    }

```

```

        Toast.makeText(this, "Deleted $deletedRows rows",
Toast.LENGTH_SHORT).show()
    }

    private fun viewData() {
        val db = dbHelper.readableDatabase

        val cursor = db?.query(
            DBContract.StudentEntry.TABLE_NAME,
            null,
            null,
            null,
            null,
            null,
            null
        )

        tvDetails.text = ""
        cursor?.moveToFirst()
        while (cursor?.moveToNext() == true) {
            val registerNumber =
cursor.getInt(cursor.getColumnIndexOrThrow(DBContract.StudentEntry.COLUMN_REGISTER_
NUMBER))
            val name =
cursor.getString(cursor.getColumnIndexOrThrow(DBContract.StudentEntry.COLUMN_NAME))
            val cgpa =
cursor.getDouble(cursor.getColumnIndexOrThrow(DBContract.StudentEntry.COLUMN_CGPA))

            tvDetails.append("Register Number: $registerNumber, Name: $name, CGPA:
$cgpa\n")
        }

        cursor?.close()
    }

    private fun clearFields() {
        etRegisterNumber.text.clear()
        etName.text.clear()
        etCGPA.text.clear()
    }

    override fun onDestroy() {
        dbHelper.close()
        super.onDestroy()
    }
}

```

DBContract.kt

```

package com.example.ex5

import android.provider.BaseColumns

class DBContract {
    class StudentEntry : BaseColumns {
        companion object {
            const val TABLE_NAME = "Student"
            const val COLUMN_REGISTER_NUMBER = "RegisterNumber"
            const val COLUMN_NAME = "Name"
            const val COLUMN_CGPA = "CGPA"
        }
    }
}

```

```
}  
}
```

UserModel.kt

```
package com.example.ex5  
  
data class UserModel(val registerNumber: Int, val name: String, val cgpa: Double)
```

UsersDBHelper.kt

```
import android.content.Context  
import android.database.sqlite.SQLiteDatabase  
import android.database.sqlite.SQLiteOpenHelper  
import com.example.ex5.DBContract  
  
class UsersDBHelper(context: Context) : SQLiteOpenHelper(context, DATABASE_NAME,  
null, DATABASE_VERSION) {  
  
    override fun onCreate(db: SQLiteDatabase) {  
        val SQL_CREATE_ENTRIES =  
            "CREATE TABLE ${DBContract.StudentEntry.TABLE_NAME} (" +  
                "${DBContract.StudentEntry.COLUMN_REGISTER_NUMBER} INTEGER  
PRIMARY KEY," +  
                "${DBContract.StudentEntry.COLUMN_NAME} TEXT," +  
                "${DBContract.StudentEntry.COLUMN_CGPA} REAL)"  
  
        db.execSQL(SQL_CREATE_ENTRIES)  
    }  
  
    override fun onUpgrade(db: SQLiteDatabase, oldVersion: Int, newVersion: Int) {  
        db.execSQL("DROP TABLE IF EXISTS ${DBContract.StudentEntry.TABLE_NAME}")  
        onCreate(db)  
    }  
  
    companion object {  
        const val DATABASE_VERSION = 1  
        const val DATABASE_NAME = "Users.db"  
    }  
}
```